A Wide Bandwidth Pulsar Timing Machine

Thesis by

José Navarro

In Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy

California Institute of Technology
Pasadena, California

1994
(Submitted May 27, 1994)

Typeset with LaTeX.

Abstract

Pulsar timing has turned millisecond pulsars into powerful tools for the study of neutron star physics, time-keeping, astrometry, the interstellar medium and binary systems. It uses millisecond pulsars as probes of the gravitational fields of their companions, of globular clusters and of the Galaxy, and provides information on the dynamics of the solar system. It has also confirmed predictions of general relativity, for which Joe Taylor and Russell Hulse received the 1993 Nobel Prize in Physics. Vasts amounts of information are expected as the current searches continue to discover new millisecond pulsars.

This thesis describes the design and construction of the Caltech Fast Pulsar Timing Machine and some preliminary results at the Parkes Observatory. The FPTM is a wide band timing machine built around a digital correlator for long-term timing stability and has a large observing bandwidth to time millisecond pulsars at high radio frequencies, where propagation effects are minimized.

It is the combination of the 128 MHz bandwidth, the 512 lag digital correlator and the 1024 period phase bins with a minimum size of 2.7 μs that gives the FPTM its power. The large bandwidth makes new observations of faint pulsars possible above 1 GHz, where current receivers easily provide radio signals with more than 100 MHz of bandwidth. The large number of channels reduces the dispersion smearing of distant millisecond pulsars, which often defines the timing floor in timing systems, and the 1024 phase bins of 2.7 μs or more ensure that the pulsar profile is adequately sampled.

The FPTM is currently at Parkes Observatory, in Australia, where we are undertaking a long-term precision timing program for millisecond pulsars. We frequently observe the bright binary millisecond pulsar J0437−4715 with sub-microsecond times-of-arrival, and have already found interesting features in the pulse profile that could not be resolved with previous timing systems. Nevertheless, we still find drifts in the timing residuals of as much as a few μs over several hours which we do not yet understand (even though they are also seen by the other timing system at Parkes!). We are confident that we will solve these residuals and consistently time PSR J0437−4715 at the level of 100 ns or better.

We also present the discovery of J0218+4232, a very luminous binary field millisecond pulsar. Its 2.3 ms pulsations were only detected after an exhaustive search that involved building new hardware.

Preface

Six years ago I visited Caltech as a prospective graduate student. I was impressed with Shri's research and resolved to read more about pulsars. I had been taught about neutron stars with superfluid interiors and exotic particle cores, and suddenly I could see pulsars as observable sources of which we catch only a glimpse every period. And it is ironically the short pulses that convey so much information about the neutron star within, and of what is between it and us.

Shri's plans to build a pulsar timing machine were starting to sound more and more interesting, indeed fascinating. The number of things we could study with such an instrument were almost too good to be true! It was going to be a large project, so I started working on it right away, with the hopes of finishing the hardware in time to get exciting scientific results before graduating.

I thought we were building the first timing machine, that we were starting a completely new field of astrophysical research. I didn't know that pulsar timing machines already existed, and I was greatly disappointed when I found that out. Indeed many discoveries had already resulted from pulsar timing, and it seemed that we were just going to make better measurements, but not study anything new.

As it turns out, however, the characteristics of the FPTM are such that we *can* study new regions of parameter space and discover new phenomena, in addition to timing pulsars more accurately. The main strength of the FPTM lies in the combination of a wide bandwidth, a large number of channels and very fine time resolution. These aspects, combined with its digital nature and resulting long-term stability, make the FPTM an unmatched precision pulsar timing system.

Acknowledgments

First and foremost, I would like to thank my advisor Shri Kulkarni for getting me involved with pulsars when I first came to visit Caltech and keeping that initial enthusiasm alive over the past six years with a constant deluge of new and exciting projects. I admire his deep knowledge of observational techniques and his ability to identify and attack a target or a project whose time has come. It's been difficult to remain sane sometimes, but it was nothing that hitting a volleyball a little harder couldn't fix. I have tried to learn as much as possible from him and hope to continue having the chance.

Building the FPTM has been an enormous project, and I would like to acknowledge several people who have made it possible. Shri himself has contributed more than his fair share of inspiration, perspiration and money. Steve Padin at OVRO has helped above and beyond the call of duty, fixing digitizers every time we managed to break one and offering advice over E-mail and the telephone when we were stranded in remote islands. Mark Hodges, Russ Keeney and Tim Clark put up with us during our visits to OVRO; Nick Scoville and Tony Readhead made the telescope and other observatory resources available to us; and John Marzano built wooden crates for the machine strong enough to survive two trips around the world and the jungles of Puerto Rico. My thanks too to Igor Lozenicins for helping out with the grunt work in the early stages, and to Reini, Roxanne, Dave Woody and the remaining people who made OVRO a home away from home.

Jagmit Sandhu has redesigned and built new modules for the FPTM, and will soon replace the dual channel IF subsystem with a quad channel system that will make dual frequency, dual polarization possible. He is inheriting the machine, and I am happy to pass along some of the successes as well as the headaches. I wish him good luck.

I am grateful to Dick Manchester, Matthew Bailes, Jonathon Bell and Andrew Lyne for their assistance at Parkes, especially Matthew, who has joined Shri, Jagmit and me deep inside the machine and has taken responsibility for the new data reduction package. I look forward to a fruitful and happy collaboration with the Parkes team in the coming years.

I have learned a lot from astronomers and engineers in other institutions. I want to thank Don Backer, Dan Werthimer, Mike Davis, Daniel Altschuler and Joe Taylor for their

encouragement years ago when I had started building the FPTM and could not yet see the light at the end of the tunnel; Dave Fort and Miller Goss for trusting me with their MkIII VLBI recorders; John Middleditch for putting up with a $90k tab in supercomputer charges; Ger de Bruyn, Dale Frail and Andrew Lyne for their collaboration in the discovery of a binary millisecond pulsar; and above all, Dale, for sharing his enthusiasm on radio astronomy and freshness and altruism in collaborations. I won't forget the freezing nights at the VLA site working with the HTRP, compiling the observing program as the telescopes slew toward the source and having it work on the first try.

Numerous trips to observatories have contributed to my happiness at Caltech, and I would like to salute and thank the friends I have made while traveling. In particular, to Norberto Despiau who made observing too much fun with his stories, don't forget Doña Panchita or else. Thanks to Willy, Willie, Angel and Rey for keeping me alive in the late night hours, to Tony for the double rations of *arroz con guinea* and to Tony Phillips for not retreating in front of a *malta grande*. I am also grateful to Marilyn and the maintenance crew, who taught me how to speak Spanish properly, before and after a bottle of *Chinchón*, and for the many games of dominoes.

I have loved Caltech from the very beginning, and will miss it. Ever since Nick, Wenge and I took Tom as an office refugee, life in the office was more fun than anywhere else except perhaps at Pie & Burger every Monday after turning in *the* problem set (don't forget that *I* was the closest to 60%). I don't think I could have found a better companion than Will for the years that we lived in the radio lab. We spent so many hours down in the basement, in front of and behind the computers! I have gained four great friends and I am sorry that we are leaving in different directions. Perhaps we'll meet upstairs in a few years...

I want to thank the people who have made life in Pasadena more exciting: my roommate Dave who builds clocks of the other kind, Candy, Sandi and Alicia, Diane and Gitta, and Helen who can find anything that's ever been written. Also Andy, Mike and Maureen, my partners in crime; Steve, for the water hole and the snakes; and Helen Johnston, for her affable personality. Gautam got closer to giving me an ulcer than Shri ever managed, but has also been a constant source of spontaneity that I appreciate. And thanks, Todd, for holding on to the rope every time I peeled, and for suggesting the trip that led to the rescue and some free gear, if not a romantic dinner.

This thesis is dedicated to my parents, for their love and support over the past twenty-nine years, and to my sisters, who came to visit me when I was too busy to visit them.

Contents

Abstract iii

Preface iv

1 Introduction 1
- 1.1 Millisecond Pulsars . . . 1
- 1.2 Timing Observations . . . 1
- 1.3 Pulsar Timing . . . 3
  - 1.3.1 Period and Period Derivative . . . 4
  - 1.3.2 Position and Frame Tie . . . 4
  - 1.3.3 Proper Motion and Parallax . . . 5
  - 1.3.4 Dispersion in the Interstellar Medium . . . 6
- 1.4 Precision Timing of Millisecond Timing . . . 7
- 1.5 A Survey of Pulsar Timing Machines . . . 8
- 1.6 The Caltech Fast Pulsar Timing Machine . . . 9
- 1.7 The New Millisecond Pulsar J0218+4232 . . . 12

2 Technical Design of the FPTM 14
- 2.1 Introduction . . . 14
- 2.2 A Historical Perspective . . . 15
- 2.3 The Data Path . . . 16
- 2.4 The Artificial Pulsar . . . 18
- 2.5 Timing and Support Signals . . . 19

3 The Millisecond Binary Pulsar J0437−4715 21
- 3.1 Introduction . . . 21
- 3.2 Current Status of the Timing Results . . . 22
- 3.3 A Short History of the Residuals . . . 23
- 3.4 Understanding the Current Residuals . . . 25
- 3.5 Discussion . . . 26

4 Discovery of a Luminous Binary Millisecond Pulsar 29
- 4.1 Introduction . . . 29
- 4.2 Observations and Analysis . . . 30
- 4.3 Discussion . . . 32

5 The IF Rack 37
- 5.1 Introduction . . . 37
- 5.2 Frequency Conversions . . . 38
  - 5.2.1 At the Telescope . . . 38

5.2.2 The Upconverter . . . 39
5.2.3 The Baseband Converter . . . 40
5.2.4 Sample Calculations of the Sky Frequency . . . 42
5.3 Power Levels . . . 45

6 The Digitizer Module **47**
6.1 Introduction . . . 47
6.2 The Digitizer ALCs . . . 47
6.3 The Digitizer Proper . . . 48
6.4 The Threshold Monitors . . . 49
6.5 Count Rates in the Digitizer Module . . . 49
6.6 Digitizer Tests . . . 51

7 The Correlator **52**
7.1 Introduction . . . 52
7.2 The Correlator Clocks . . . 53
7.3 The Correlator Chip . . . 54
7.4 The JPL Accumulator Chip . . . 56
7.5 Correlator Rates . . . 57
7.6 Clock and Data Issues . . . 58

8 The Pulsar Backend **60**
8.1 Introduction . . . 60
8.2 Folding at the Pulsar Period . . . 62
8.3 The Ramadder Boards . . . 62
8.3.1 The RAW Cycles . . . 64
8.3.2 The Maximum Phase Bin Size . . . 65
8.3.3 The Output Memories . . . 66
8.3.4 Count Rates in the Ramadder Boards . . . 67
8.3.5 Cross Talk and Double Buffering . . . 68
8.4 The Control Board . . . 69
8.4.1 Generating the Pulsar Spin Frequency . . . 71
8.4.2 The Start of an Observation . . . 71
8.4.3 Intrinsic Timing Accuracy . . . 72
8.5 Clock and Data Issues . . . 72

A The VMEbus **74**
A.1 Overview . . . 74
A.1.1 VMEbus Formats . . . 74
A.1.2 Termination . . . 75
A.1.3 Master and Slave Interfaces . . . 78
A.1.4 Read and Write Cycles . . . 78
A.1.5 Interrupts . . . 79
A.2 The Technical VMEbus . . . 79
A.2.1 Read and Write Cycles Revisited . . . 80
A.2.2 Interrupts Revisited . . . 81
A.3 The FPTM VMEbus Interface . . . 82

B The PIO Board **84**
B.1 Introduction . . . 84
B.2 Receiving Data . . . 84
B.3 Sending Data . . . 85
B.4 Addressing the PIO Board . . . 85

B.5 The Status Register 86
B.6 History and Particulars 87

C The Digital Interface **89**
C.1 Introduction 89
C.2 Data Flow Overview 90
C.3 Technical Description 90
C.4 History of the Digital Interface 93

D The Buffer Board **94**
D.1 The Front-Panel Input 94
D.2 The VMEbus Output 95
D.3 Memory Switching 95
D.4 Selftest Features 95
D.5 Internal Registers 96
D.6 Internal Functions 97
D.7 CPU Control of the Buffer Board 97
D.8 History 98
D.9 Modifications to the Rev.A PCB 99

E The FPTM at Parkes **100**
E.1 Introduction 100
E.2 AC Power 100
E.3 Signal Termination 100
E.4 Power Levels 101
E.5 Voltage Levels 101
E.6 The Artificial Pulsar Generator 102

F Glossary **103**

Bibliography **106**

List of Figures

1.1 Synchronous Folding of the Pulsar Signal 2

2.1 A Block Diagram of the FPTM 17

3.1 Single Night Timing Residuals With the FPTM 22
3.2 A Comparison of Residuals From the FPTM and the Filterbank 23
3.3 The Pulse Profile of J0437−4715 27

4.1 P-band Images of PSR J0218+4232 31
4.2 Orbital Period Fit for J0218+4232 33

5.1 Downconversion of the 2.4 GHz Band 40
5.2 The IF Subsystem 41

7.1 The Heart of the Correlator 53
7.2 The Correlator Outputs 57
7.3 Clock and Data Delays 59

8.1 From Correlator Data to Pulse Profiles 61
8.2 A Ramadder Board Section 63
8.3 Control Sequence for a RAW Burst 65
8.4 A Single RAW Cycle 66
8.5 The Signals in a Ramadder Board Section 69

List of Tables

1.1 Comparison of Timing Systems 10

4.1 Parameters of PSR J0218+4232 35
4.2 Parameters of Field Low-Mass Binary Pulsars 36

5.1 IF Configuration at Parkes 39
5.2 Sample Setups for the IF Chain 43
5.3 Signal Levels 45

6.1 Digitizer Thresholds and Output Values 49
6.2 Expected Threshold Monitor and Digitizer Rates 51

7.1 Correlator Clocks and Bandwidth 54
7.2 Multiplication Table 55

8.1 Control Signals for the Ramadder Boards 70

A.1 VME Board Dimensions 75
A.2 VMEbus P1 Layout 76
A.3 VMEbus P2 Layout 77
A.4 Address Modifier Bits 80
A.5 Data Transfers Over the VMEbus 81

B.1 PIO Board Addressing 86
B.2 PIO Board Status Byte 86

C.1 An Example 91
C.2 Transfers Between the PIO Board and the Middle Board 92
C.3 Transfers Between the Middle Board and the End Board 92

D.1 The Control Register 96
D.2 Internal Functions of the Buffer Board 97
D.3 Buffer Board Addressing 97

E.1 AC Power Budget 101

F.1 Glossary 103

Chapter 1

Introduction

1.1 Millisecond Pulsars

More than 600 pulsars have been found since PSR B1919+21 was discovered in 1967 by Jocelyn Bell and Anthony Hewish (Hewish *et al.* 1968). In 1982 a new class of pulsars was revealed with the discovery of the first millisecond pulsar B1937+21 (Backer *et al.* 1982). They have short periods ($\leq$20 ms) and small magnetic field strengths ($\leq 10^9$ G). We believe that millisecond pulsars are spun up to short periods by accretion of matter from a companion. Indeed many of them are in binary systems with low-mass companions (see Bhattacharya and van den Heuvel 1991 and references therein).

Millisecond pulsars are even better clocks than ordinary pulsars, with rotational stabilities in some cases as small as 1 part in 10^{14} (Kaspi, Taylor, and Ryba 1994). This, combined with the fact that their short periods make it possible to measure the pulse times-of-arrival (TOAs) to great accuracy, makes them attractive targets for precision timing observations and the subject of numerous searches (see the review in Camilo 1994).

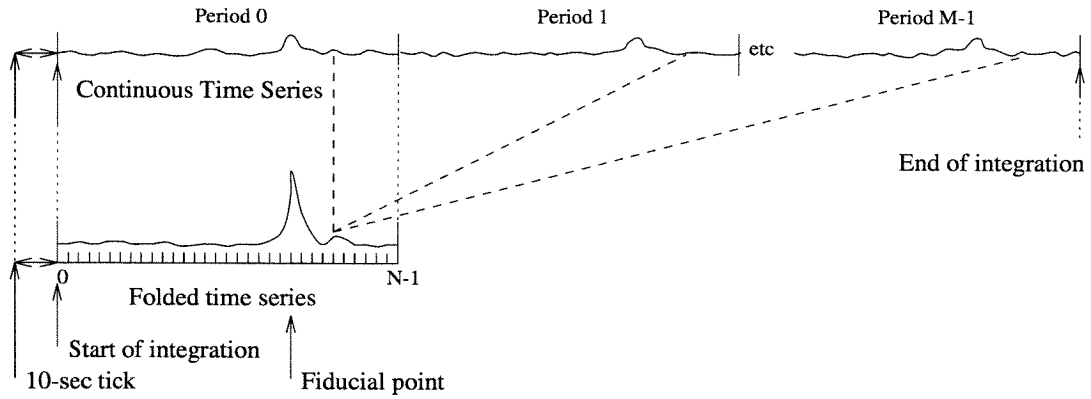
In this chapter we review the basic principles of pulsar timing, the main scientific objectives of millisecond pulsar timing, and the current state of the art of timing pulsars.

1.2 Timing Observations

To "time a pulsar" is to record the times of arrival of its pulses. A time-of-arrival, or TOA, is a time tag that we attach to one of the pulses from the pulsar. It is not necessary to measure a TOA for every pulse that we receive, and indeed in most cases single pulses are too weak to be detected above the noise. Instead, one observes for a few minutes and obtains one global TOA that corresponds to a pulse near the middle of the integration.

The observations involve folding the pulsar signal synchronously with the pulsar period, so that one ends up with a single pulse profile per frequency channel (figure 1.1). The TOA of that integrated, single pulse is the sum of the start time of the observation and the offset

Figure 1.1: Synchronous Folding of the Pulsar Signal
The data from each channel is folded synchronously with the pulsar period into a number of bins (N). The pulse arrival time from that integration is obtained from the start time (the time of the 10-sec tick plus a small, measurable offset) and the offset to the fiducial point in the profile. The location of the fiducial point is found by crosscorrelation of the profile with a standard template.



between the first bin of the pulse profile and the pulse peak. To minimize the effects of errors in the pulsar model, the TOA is normally referred to a period near the middle of the integration instead of the beginning.

A list of TOAs and observing frequencies is obtained after a series of observations, from which a better pulsar model can be derived by means of the TEMPO package (Taylor *et al.* 1976). TEMPO transfers the topocentric TOA to the solar system barycenter and removes the pulsar's binary orbit if necessary, after which the TOAs should be exact multiples of the pulsar period, with a small growing delay due to the pulsar spin-down. The parameters of the timing model can be kept constant or fit to the input TOAs, depending on the nature of the data. For example, the pulsar dispersion measure can be easily obtained from dual frequency observations, but is best left fixed if the current TOAs are all at the same frequency and cannot provide any new information on the dispersion measure.

The precision of a TOA is given roughly by the pulse width divided by the signal-to-noise ratio of the observation. For the pulsar J0437−4715 a typical one-minute observation at 1500 MHz has an SNR of ∼100, which with a main pulse width of 0.12 ms gives a TOA error of ∼1.2 μs. Longer integrations reduce the TOA error even further, but it is preferable to start a new observation as soon as the single-point TOA gets to an estimated precision near 100 ns, to get an independent timing point and check the internal consistency of the pulsar timing system.

1.3 Pulsar Timing

The pulse arrival time t_b at the solar system barycenter (SSB) can be derived from the topocentric arrival time t from (Taylor and Weisberg 1989)

$$t_b = t + (\mathbf{r} \cdot \hat{\mathbf{n}})/c - D/f_b^2 + \Delta_{E\odot} - \Delta_{S\odot}. \tag{1.1}$$

In this equation t_b is measured at the barycenter in units of barycentric dynamical time (TDB) and t is measured at the topocenter in terrestrial dynamical time (TDT). $\hat{\mathbf{n}}$ is a unit vector toward the pulsar, $\mathbf{r}$ is a vector from the SSB to the phase center of the telescope at the time of observation, and c is the speed of light. $\Delta_{E\odot}$ is the solar system "Einstein delay," the combined effect of gravitational redshift and time dilation due to Earth's motion, and $\Delta_{S\odot}$ is the "Shapiro delay" caused by propagation of the pulsar signal through curved spacetime near the Sun (Shapiro 1964).

The term D/f_b^2 removes the dispersive delay and converts the arrival time to infinite frequency. The observing frequency f_b is expressed in the rest frame of the SSB, and the dispersion constant D is given by

$$D = \frac{e^2}{2\pi mc}\int_0^d n_e dl \equiv \frac{\mathrm{DM(cm^{-3}pc)}}{2.41\times 10^{-16}}\ \mathrm{s}^{-1}, \tag{1.2}$$

where DM is the "dispersion measure" due to the column density of free electrons toward the pulsar.

The conversion of t to t_b requires the pulsar position and a model of the solar system, but it is the pulse *phase* as a function of barycentric time that conveys the remaining information about the pulsar. The pulse phase $\phi(t_b)$ at any given time is given by

$$\phi(t) = \phi_0 + \nu_0(t - t_0) + \frac{1}{2}\dot{\nu}(t - t_0)^2 + \frac{1}{6}\ddot{\nu}(t - t_0)^3 + \ldots \tag{1.3}$$

where ϕ_0 is the pulse phase at $t = t_0$, and $\nu(t) = 1/P(t)$ is the (observed) spin frequency of the pulsar and $\dot{\nu}$, $\ddot{\nu}$ its derivatives with respect to time. The subscript on barycentric times has been dropped for clarity.

The spin frequency ν and its time derivatives are relatively simple functions of time in the case of isolated pulsars (Manchester and Taylor 1977), but can be extremely complicated for binary systems. We will look at this again in the next section, where we explore the applications of pulsar timing in terms of equations (1.1)–(1.3).

The scientific applications of precision pulsar timing are many, and have been extensively reviewed in the literature (see Backer and Hellings 1986, Ryba 1991 and references therein). We now briefly explain how each application is derived from the TOAs at the barycenter and the magnitude of each effect.

1.3.1 Period and Period Derivative

The periods of some millisecond pulsars are known to 1 part in 10^{14} because the individual TOAs can be measured to $\sim$1 μs over many years, and 1 μs/yr = 3.2×10^{-14}. Period derivatives as small as 10^{-25} s/s can also be measured. In the case of J0437−4715, which we use as an example, the integrated delay on the pulse arrival times over one year of observations due to $\dot{P}$ would be, from equation (1.3):

$$-\Delta\phi(t) \approx -\frac{1}{2}\,\dot{\nu}\,t^2 = \frac{1}{4}\,\frac{\dot{P}}{P^2}\,t^2 = \frac{1}{4}\,\frac{1.2\times10^{-19}}{(5.7\text{ ms})^2}\,(1\text{ yr})^2 = 3.9\text{ periods}, \qquad (1.4)$$

or 22.6 ms in one year. This delay is easily observable.

There are several terms that can contribute to the apparent $\dot{P}$: the intrinsic $\dot{P}$ itself, which is always positive and has values between 10^{-12} and 10^{-20} or smaller; the Shklovskii effect due to transverse motion (Shklovskii 1970), which can be significant for fast, nearby pulsars and perhaps the closest known pulsar PSR J0108-1431 (Tauris *et al.* 1994); and any acceleration caused on the pulsar by strong gravitational fields such as those felt by globular cluster pulsars.

The Shklovskii term has been measured for a few millisecond pulsars (Camilo, Thorsett, and Kulkarni 1993). Once its effect is removed from the apparent $\dot{P}$, the intrinsic $\dot{P}$ and the intrinsic characteristic age $\tau = P/2\dot{P}$ are known. This effect should be observable in most of the new millisecond pulsars that have been recently discovered and not yet timed.

Some globular cluster pulsars present negative apparent period derivatives. Both the intrinsic $\dot{P}$ and the Shklovskii $\dot{P}$ are positive, so any measured negative $\dot{P}$ must be due to acceleration by the cluster, and gives a lower limit to that acceleration. Phinney 1992 has used the measured negative $\dot{P}$ to explore the gravitational potentials of clusters.

Equation 1.4 reflects one of the properties of pulsar timing: that the timing residuals produced by an incorrect model can increase as a nonlinear function of time. As a result, while the linear ones like period will improve as $t^{3/2}$ (a factor of t for time and a factor of $t^{1/2}$ for the increase in SNR), some of the pulsar parameters will become much more accurate with time. Other parameters, like DM, only get better as $t^{1/2}$ because the only contributing factor is the *number* of observations.

1.3.2 Position and Frame Tie

Errors in the pulsar position result in sinusoidal timing residuals with a one year periodicity, the amplitude of which depends on where the pulsar is with respect to the ecliptic plane and can be expressed as

$$[\Delta t]^2 = [4.8\ \mu\text{s}\ \cos\lambda\ \Delta\beta]^2 + [4.8\ \mu\text{s}\ \sin\lambda\ \Delta\lambda]^2, \qquad (1.5)$$

where β and λ are the ecliptic longitude and latitude and their errors are given in milliarcseconds. Several years of timing can produce extremely accurate pulsar positions (0.1 mas for B1937+21 and B1855+09 in Kaspi, Taylor, and Ryba 1994), sometimes better than the VLA or VLBI positions (and needing only a single radio dish). The timing solution is in ecliptic coordinates, while the interferometric position is in equatorial coordinates because it is based on Earth's rotation. The combination of the two positions ties the two coordinate systems and is a step toward what is called "reference frame tie."

Proper frame tie requires the positions of several pulsars distributed all over the sky to be determined very accurately with *both* timing and interferometry. The difficulty in improving the frame tie is that there are few pulsars that can be timed to positions better than 0.1 mas with both methods, as they must be fast for timing and bright for interferometry. The current pulsar searches are improving the prospects of pulsar astrometry by finding many new millisecond pulsars.

1.3.3 Proper Motion and Parallax

A model error in the proper motion of a pulsar is equivalent to a position error that changes with time and produces annual sinusoidal residuals whose amplitude grows linearly with time. If we rewrite equation (1.5) as

$$\Delta t \approx 4.8\ \mu\mathrm{s}\ \Delta\rho, \qquad [\Delta\rho]^2 = [\cos\lambda\ \Delta\beta]^2 + [\sin\lambda\ \Delta\lambda]^2,$$

where $\Delta\rho$ is a measure of the position error, still in milliarcseconds, then an error in the proper motion will correspond to a timing residual of

$$\Delta t = 4.8\ \mu\mathrm{s}\ \Delta\rho = 2.4\ \mu\ \frac{\mu\mathrm{s}}{\mathrm{yr}} = 2.4\ V_{10}\ d_{\mathrm{kpc}}^{-1}\ \frac{\mu\mathrm{s}}{\mathrm{yr}}, \tag{1.6}$$

where the proper motion μ is given in mas/yr, and where V_{10} is the transverse velocity in units of 10 km/s and d_{kpc} is the distance to the pulsar in kiloparsecs.

The distance to the pulsar can be measured through parallax, since parallax will also produce annual sinusoidal residuals:

$$\Delta t = 1.2\ \cos\lambda\ d_{\mathrm{kpc}}^{-1}\ \mu\mathrm{s}. \tag{1.7}$$

Parallax is the only direct way to obtain the distance to a pulsar, and is instrumental to understanding the electron distribution in the local insterstellar medium. For example, the parallax measurements of PSR B1451−68 improved the estimate of the electron density in the direction of the pulsar from 0.029±0.014 cm^{-3} (Salter, Lyne, and Anderson 1979) to 0.019±0.003 cm^{-3} (Bailes *et al.* 1990).

Proper motion, when measured simultaneously with pulsar timing and interferometry,

is useful for understanding the relative orientation and rotation of the different solar system ephemerides in use. If combined with parallax measurements or more approximate distance estimators such as from interstellar scattering observations, proper motion gives the transverse velocity of the pulsar (eq. 1.6), which can be used to constrain pulsar formation scenarios (Phinney and Kulkarni 1994).

1.3.4 Dispersion in the Interstellar Medium

The term $-D/f^2$ in equation (1.1) is a correction for the dispersion of the radio signals in the interstellar medium. This term is only quadratic, however, and does not correct the delays due to the interstellar medium perfectly. We can obtain a better estimate of the delay by deriving it from its source, the frequency-dependent group velocity of radio waves in a cold plasma (Manchester and Taylor 1977)

$$v_g = c\,(1-\omega_p^2/\omega^2)^{1/2}, \qquad \omega_p^2 = \frac{4\pi n_e e^2}{m_e}, \tag{1.8}$$

from which we can calculate the travel time for a wave of frequency $f=\omega/2\pi$:

$$t = \int_0^d v_g dl \approx \frac{1}{c}\int_0^d dl + \frac{e^2}{2\pi m_e c}\int_0^d n_e dl\,\frac{1}{f^2} + \frac{3e^4}{8\pi^2 m_e^2 c}\int_0^d n_e^2 dl\,\frac{1}{f^4}$$

$$= \frac{d}{c} + \frac{\mathrm{DM(pc\ cm^{-3})}}{2.41\times10^{-16}}\,\frac{1}{f^2} + \frac{\mathrm{EM(pc\ cm^{-6})}}{4.00\times10^{-24}}\,\frac{1}{f^4}$$

where DM is the "dispersion measure" and EM is the "emission measure." If the plasma is magnetized, there will be a term depending on RM/f^3, where RM is the rotation measure and depends on the magnetic field along the line-of-sight. From Phillips (1992),

$$t = \frac{d}{c} + \frac{\mathrm{DM(pc\ cm^{-3})}}{2.41\times10^{-16}}\,\frac{1}{f^2} + p\,\frac{\mathrm{RM(rad\ m^{-2})}}{1.75\times10^{-17}}\,\frac{1}{f^3} + \frac{\mathrm{EM(pc\ cm^{-6})}}{4.00\times10^{-24}}\,\frac{1}{f^4}$$

where p is $+1$ for RCP and -1 for LCP. The cubic and quartic terms are usually negligible except at very low frequencies $f<100$ MHz.

What is more important than the f^{-3} and f^{-4} terms is the time variability of the f^{-2} term, the dispersion term, which produces time-variable delays and residuals. One way to avoid contamination of the timing residuals from changes in dispersion is to measure the DM during every observation. This is impractical because it requires observations at two frequencies with the associated receiver change. Alternative, one can observe at very high frequencies alone, where the variations become smaller. The best solution altogether is to observe at *two* high frequencies to obtain the DM and to be immune from the other propagation effects. This is exactly what Kaspi, Taylor and Ryba (1994) do for PSR B1937+21. They alternate measurements at 1.4 and 2.4 GHz, and measure DM drifts

of 0.001 pc cm^{-3} per year, which are equivalent to 1.5 μs/yr at a frequency of 1.4 GHz. It is clear that observations at two frequencies are necessary for submicrosecond timing.

1.4 Precision Timing of Millisecond Timing

So far we have discussed the basic applications of pulsar timing. When the pulsar is in a binary system, the delays due to the motion of the pulsar within the binary orbit must be considered. In tight binaries with massive companions, general relativistic effects can be very important. Indeed this is a fruitful field of astronomy, and recently Joe Taylor and Russell Hulse were awarded the Nobel Prize for the discovery of the eccentric binary pulsar B1913+16, with which Taylor and co-workers were able to verify the radiative predictions of general relativity.

A number of new applications emerge from precision timing when the times-of-arrival can be measured at the microsecond level. As explained above, millisecond pulsars are ideal for these applications because of their short periods and inherent stability. Examples of these applications are the following:

a) The use of TOAs to constrain or perhaps even detect a cosmic gravitational wave background (akin to the 3 K photon background). In the wavelength regime of 10^6–10^8 s, the most sensitive method presently available for the detection of a stochastic background is timing of millisecond pulsars (Stinebring *et al.* 1990). The lower limit of 10^6 s is set by the integration time needed to beat down the residuals, and the upper limit of 10^8 s is determined by the uncertainties in the solar system ephemerides, which are needed to evaluate the large term $(\mathbf{r} \cdot \hat{\mathbf{n}})/c$ in equation (1.1).

b) The determination pulsar binary parameters including the masses of the companion, as in PSR B1855+09 (Thorsett *et al.* 1993). A similar and important application comes from the fact that in low mass systems, like J0437−4715, we do not expect any significant contribution to the orbital period derivative $\dot{P}_b$ due to general relativity. The measured $\dot{P}_b$ can be attributed entirely to the differential acceleration caused by the Galactic disk on the pulsar system and on the solar system, and can yield a unique constraint on the mass distribution of our Galactic disk.

c) The determination of parallax to probe the large scale structure of the interstellar plasma. Proper motion measurements lead to space velocities which are fundamental toward understanding the formation of millisecond pulsars.

d) Multi-frequency pulsar timing to probe the large scale structure of the interstellar plasma. The pulsar signal is scattered by irregularities in the ionized interstellar medium, which results in scatter broadening of the pulse features, and in stochastic shifts in the TOAs as the irregularities drift across our line-of-sight. Millisecond pulsars can be used to measure these shifts, and from the observed residuals one can derive the power spectrum

of the irregularities. Foster and Cordes (1990) discuss the limitations to pulsar timing due to interstellar scattering.

The ultimate limit to pulsar timing is naturally the pulsar itself. However, as stated earlier, it appears that millisecond pulsars, especially those with old ages ($>10^9$ yr, e.g., B1855+09) are essentially noiseless. If so, the limitations to precision timing are propagation in an irregular ISM, radiometric noise in the telescope receiver, and the pulsar timing backend. The first limitation can be overcome by timing pulsars at high frequencies where the scattering effects caused by the interstellar plasma become very small, and at two different frequencies if possible to remove all trace of variations in dispersion measure. The second limitation can be overcome with a timing machine capable of processing wide band signals, since radiometric noise is proportional to $B^{-1/2}$. Finally, the pulsar timing backend should be able to process the wide band signals and have many spectral channels and high time resolution. A fully digital system is favored since all the applications discussed above require long term stability.

All these considerations led us to the concept of a wide band digital correlator system for pulsar timing, specifically to be used to time pulsars at the highest possible frequency.

1.5 A Survey of Pulsar Timing Machines

Interstellar dispersion produces a frequency-dependent delay in the pulsar's signal. The detection of the signal collapses all the delays with the effective result of smearing the pulse profile. In the case of millisecond pulsars, the smearing can be so large as to completely wash out the pulsations and render the observation useless. The dispersion is reversible, however, if one can delay the different frequencies in the observing bandwidth by different amounts before adding them. This leads to several techniques for dedispersion that can be classified into two groups: coherent and incoherent dedispersion, also called pre-detection and post-detection dedispersion.

Coherent dedispersion applies a transfer function to the radio waves that exactly counters the dispersion from the interstellar medium. After coherent dedispersion one still possesses a radio signal that can be upconverted, downconverted, filtered or detected. It is the ideal way to dedisperse pulsar signals, but is difficult to do because of the extreme computational requirements. So far it has only been used in small bandwidth timing systems, and with a limitation to small dispersion measures. Don Backer at Berkeley is constructing a large bandwidth coherent dedispersion pulsar timing machine that uses custom VLSI technology.

Post-detection or incoherent techniques rely on passing the radio signal through a filterbank, detecting each channel separately and then adding them all with appropriate delays. These techniques are normally preferred over coherent dedispersion because filterbanks are

relatively easy and inexpensive to make. However, filterbanks are analog devices and the passband shapes are variable from filter to filter and can change with time. The individual filter shapes must be measured regularly to account for the systematic delays introduced by scintillation through gradients of intensity as a function of observing frequency. As a result, filterbanks may lack long-term stability, which makes them potentially undesirable for long-term precision pulsar timing. An alternative to working in the frequency domain is offered by the correlator approach. A correlator-based pulsar timing system folds the autocorrelation (or crosscorrelation) function of the incoming signal at the pulsar period, instead of the individual frequency channels.

Table 1.1 shows the characteristics of the pulsar timing machines in use today. The highest time resolution is achieved with pre-detection systems, but they are limited to small bandwidths (except for the Berkeley CDRP, which will come on-line in the next few years). On the other hand, post-detection systems have larger bandwidths but are limited by their sampling rates and the relatively small number of channels.

1.6 The Caltech Fast Pulsar Timing Machine

The Caltech FPTM is a precision pulsar timing machine built around a wide band digital correlator. Its primary characteristics are a maximum input bandwidth of 128 MHz, a large number of channels (512 total) and excellent time resolution (a minimum of 2.7 μs per phase bin). The 512 channels are actually 512 lags in the correlator, which can be converted into frequency channels prior to dedispersion during data reduction.

The FPTM correlator is very flexible and its 512 lags can be configured in many ways in groups of 64. The standard configurations are 2×256 lags for pure timing observations and 4×128 lags for polarization observations. More specifically, if L and R are the IF signals corresponding to the left and right circular polarizations, then standard timing observations use 256 lags for LL and 256 for RR to record only intensity in each polarization. On the other hand, polarization observations use 128 lags for each of LL, RR, LR and RL, from which the Stokes parameters I, V, Q, and U can be obtained.

The idea of a building a pulsar timing machine arose with the OVRO Millimeter Correlator. In most correlators the outputs of the multipliers are accumulated in the same chip that does the multiplications, and data are read out at most once every few milliseconds (the phase switching period in some interferometers). As a result, the time resolution of these correlators is poor, and cannot be used for millisecond pulsar timing. The OVRO correlator is different because the accumulation is done off-chip. Hence there is a point on the correlator boards between the correlator chips and the accumulator chips that we can tap to perform our own specialized, pulsar synchronous accumulation.

The Caltech Fast Pulsar Timing Machine was designed and built around the mm-array

Table 1.1: Comparison of Timing Systems

A comparison of the existing timing systems. The configurations listed here are those using the maximum bandwidth in two polarizations, with no sampling of the cross-polarizations. Maximum bandwidths are in MHz and minimum sampling times are in μs. N_{chan} is the number of channels across the entire bandwidth and N_{bins}, the number of phase bins across and entire pulse period. For short period pulsars N_{bins} is effectively decreased because the minimum phase bin size is Δt_{min}. The number of phase bins for a 1.5 ms pulsar is given in the next column. The *type* column lists the method used to break the spectrum into channels, and the *pol* column lists whether cross-polarization observations are possible.

FPTM: Caltech Fast Pulsar Timing Machine at Parkes; MK3-VLA: at the Very Large Array (Thorsett 1991); MK3-AO: Princeton Mark III system at Arecibo (Ryba 1991); MK3-GB: at Green Bank 85 ft (Stinebring *et al.* 1992); SP: Spectral Processor at Green Bank 140 ft (Fisher 1989); SP': New Spectral Processor (Fisher 1994); FPSM: Berkeley Fast Pulsar Search Machine at Arecibo (Backer *et al.* 1990); HTRP: High Time Resolution Processor at the VLA (McKinnon 1992); PKS-FB: Parkes Filterbank System, a search machine that can be used for timing (Johnston *et al.* 1993); AOC: Arecibo Correlator (Hagen 1987); MK3-CD: Princeton Mark III Coherent Dedispersion System (Ryba 1991); NSLO: Nançay Swept Local Oscillator timing machine at Nançay (Lestrade *et al.* 1990); CDRP: Berkeley Coherent Dispersion Removal Processor (Backer *et al.* 1994), which will be installed at the Green Bank 140 ft telescope.

System	BW_{max}	Δt_{min}	N_{chan}	N_{bins}	$N_{bins}^{1.5\ \mathrm{ms}}$	Type	Pol
FPTM	128	2.7	256	1024	561	corr	y
MK3-VLA	46	11.4	14	1024	131	filter	y
MK3-AO	40	11.4	16	1024	131	filter	y
MK3-GB	16	11.4	16	1024	131	filter	y
SP	20	12.8	256	128	117	fft	n
SP'	10	25.6	512	64	58	fft	n
FPSM	40	25.0	128	1024	60	corr	n
HTRP	28	140.0	14	1024	10	filter	y
PKS-FB	320	150.0	64	–	10	filter	y
AOC	20	500.0	128	1024	3	corr	y
MK3-CD	4	1.4	–	8192	1071	pre-det	y
NSLO	10	1.2	–	2500	1250	pre-det	n
CDRP	64	1.0	–	1024	1024	pre-det	y

correlator chip. For this I had to clone one baseline of the correlator, which involved duplicating 8 correlator boards and all the support hardware. The correlator boards were especially challenging because of the high speed ECL signals, which at 256 MHz are very difficult to handle. I had to learn them inside out, and in the process of testing and debugging my copies of the boards I found and fixed some "bugs." The correlator boards are described in chapter 7.

The most important bug involved some swapped clocks, and was not important at low clock rates but prevented the boards from functioning above 220 MHz. The fix was to swap 18 pairs of signals per board by means of a socket-to-socket patch underneath the JPL accumulator chips. This happened at the same time that OVRO was putting together its first three correlator baselines, and they were able to take advantage of my modifications and observe with bandwidths larger than 110 MHz. The changes have now been incorporated into the printed circuit correlator boards (PCBs) that were built for the expansion of the mm-array to more than 3 antennas. The correlator boards had other problems, most of which were not fundamental but made operation more difficult. I found some design errors in the bus readout circuitry, that happened to work by chance. My changes were also incorporated in the new PCB version of the correlator boards.

I learned a lot about radio astronomy instrumentation when duplicating the modules associated with correlator, especially the digitizer and the clock generation and distribution modules, and the IF subsystem. While debugging the IF chain, starting with a noise source and ending with the digitizer, I found a bug in the digitizer which Steve Padin had independently found and fixed in the OVRO versions. Chapters 5 and 6 describe the IF subsystem and the digitizer.

Instead of a using a commercial VMEbus data I/O board to exchange data (and programs) between the correlator and control computers, I built my own digital I/O boards. This allowed me to give the PIO boards the features I wanted: choice of polled or interrupt driven, parallel digital buffered data and simultaneous input and output. The PIO board is described in appendix B. Learning the VMEbus well enough to design my own VMEbus boards was a painful process, and I summarize the essentials of the bus in appendix A to provide a reference more basic than the existing literature.

The PIO board is also used for the digital interface, a fan-out box with several modules that can be inserted into other boxes to provide computer control. Each control module contains two 32-bit registers that can be used in a synchronous or asynchronous way, and with two control lines per register for the instrument to be controlled. I designed the digital interface after realizing that having to set all the front panel switches by hand to configure the FPTM is impractical, especially if when trying to do remote observing. The digital interface gives the control computer complete power to configure the FPTM from the software. It is described in appendix C.

Most of the design effort went into the pulsar backend. From the hardware point of view, it involved building the ramadder boards and the control board. The ramadder boards are described in chapter 8 and were the most challenging, partly because I designed them early on when I had less experience, but also because they are quite complex *per se*. Each board must fold 2 Gbit of data *per second* from the associated correlator board, and it can only do so by being very parallel. The time resolution of 2.7 μs per phase bin was hard to achieve, but it perfectly complements the qualities of the correlator for millisecond pulsar timing, as can be seen in table 1.1. It also made the control board harder to design, as the 60+ control signals that it provides to the ramadder boards have to be properly phased.

The software effort that went into the pulsar backend is considerable. After much consideration I decided to use OS9, a real-time operating system that looks much like Unix and runs in embedded computers. The observing programs are written in OS9/C and the data reduction package is in Unix/C. Part of the data reduction package was written by Will Deich (Deich 1994), but I had to provide interface for the FPTM data. The correlator computer also runs OS9, but off a ROM chip because it does not have a hard disk. The test and setup programs for the correlator are loaded into the correlator RAM via the PIO boards.

To date, the FPTM has observed pulsars at OVRO, Arecibo and Parkes. Each observatory has supplied a number of challenges, perhaps the biggest of which were the ground loops at Arecibo. Things are not good when plugging a BNC cable into the machine means a shower of sparks, or when lightning destroys your hard disk. However, each observing trip has been equally rewarding both on a personal level and from an astronomical point of view, and I hope they continue to be. Chapter 2.5 shows some of the results from Parkes.

1.7 The New Millisecond Pulsar J0218+4232

The original plan for my thesis was to search for new millisecond pulsars and simultaneously build the FPTM to time them. I started a pulsar search from Arecibo, but it was not very successful, and soon I was fully involved in the timing machine and had no time to continue with undirected searches.

In 1991 we serendipitously discovered a highly polarized, compact radio source that had all the looks of a pulsar, and started a pulse search from the VLA. The fact that we didn't see any pulsations made the search all the more interesting, for now the pulsar had to be very fast or not at all, and this side project took a new intensity.

Unfortunately, this source lies just 4° North of the Arecibo declination range, and the pulsar search hardware available at the VLA was inadequate. We decided to try a completely new approach, for which I designed a new digital board to interface an OS9

computer to a MkIII VLBI recorder, and replay the MkIII baseband signals through the OS9 system an onto a Unix workstation where they could be analyzed. We eventually discovered the source to be a 2.3 ms pulsar in a 2 day binary orbit around a 0.15 $M_\odot$ star. It is probably the most luminous known millisecond pulsar and it is presented in chapter 4.

Chapter 2

Technical Design of the FPTM

2.1 Introduction

The Fast Pulsar Timing Machine is a correlator-based pulsar timing machine with a maximum bandwidth of 128 MHz. It has an IF subsystem that operates in the range 0–1 GHz and converts the radio signals down to baseband, filters, a dual digitizer, a digital correlator and a pulsar backend that averages the correlator output.

The digital correlator is highly modular and can be configured in many ways. What is described here is the Parkes configuration of early 1994, with a dual IF subsystem, a fast digitizer, possible bandwidths of 2, 8, 32 and 128 MHz and a maximum of 512 lags. In this configuration, the 512 lags of the correlator can be organized for autocorrelation or crosscorrelation of the two input polarizations, in groups of 64 lags, and the lag data can be folded at the pulsar period into at most 1024 phase bins, with a minimum bin size of 2.7 μs.

All the basic modules of the FPTM are described in detail in chapters 5–8 and appendices A–F. Many of the modules were cloned from the Owens Valley Millimeter Array, in particular the baseband converters, the digitizer, the clock generator, the clock distribution module and the correlator. Some of these we modified to suit our needs and repackaged in a more convenient form for the FPTM. Some we redesigned and improved on, like the clock generator and the clock distribution module, which we converted from gallium arsenide to ECLps technology when the new generation silicon chips first appeared. The pulsar backend was designed and built from scratch, with the hardware for communications and computer control, the artificial pulsar, and the software to test and run the entire timing machine.

The strength of the machine lies in the combination of all its parts: a wide bandwidth IF subsystem, a fast digitizer, a fast digital correlator with a large number of lags, and a pulsar backend with fine time resolution. In addition, the fact that it can easily be configured through software makes testing and observing easier. A sophisticated software

package already exists to analyze the timing data (Deich 1994), which we are extending to include polarization analysis and other routines specific for the FPTM.

2.2 A Historical Perspective

The idea of building a new pulsar timing machine occurred to my advisor Shri Kulkarni in 1987, at the same time that the Owens Valley Radio Observatory started to develop a large bandwidth digital correlator for its Millimeter Array (Padin *et al.* 1993). M. S. Ewing was designing the high speed correlator chip, and was upset that he could not fit enough bits of accumulation inside the chip itself for lack of available gates. The limitations of ECL gate array technology at that time turned out to be a blessing for the timing machine project, for now external accumulation was necessary and it could be done in a pulsar synchronous manner (§7.3). Had it been possible to accumulate further in the correlator chip, it is very likely that the outputs would have been too slow to time millisecond pulsars.

Marty Ewing, having worked on pulsars in the past (Filippenko, Readhead, and Ewing 1983), was also interested in the possibility of using the correlator for pulsar timing, and laid out the correlator boards with traces and connectors to make off-board accumulation possible. All we had to do was connect ribbon cables to the correlator boards and design a backend that would process the correlator data. No easy task, though, as the total output from the correlator amounts to no less than 2 Gbyte per second.

In 1988 we submitted a President's Fund Proposal for seed money to build the pulsar backend. The original proposal called for a 256 lag timing machine with 256 phase bins per period, and a double buffering scheme for reading out the folded data without interrupting the observations. This proposal was approved and we started to design and build the timing machine, adding on its capabilities when possible, until it became a 512 lag, 1024 phase bin monster. It currently weighs 1500 lbs!

The correlator requires a series of other modules in order to function, and with the help of the OVRO staff, we cloned most of those from the OVRO millimeter correlator project, mainly the baseband converter, digitizer, clock generator and clock distribution module. At the same time we gave the millimeter correlator project some feedback, helping them debug the correlator boards, and later by designing ECLps versions of the clock distribution box and the clock generator.

The FPTM was mostly designed at Caltech in the Radio Lab (Robinson 004) and was put together at OVRO, where it saw first noise (the correlator equivalent of optical first light) on October 8, 1992. The first astronomical observations were done at Arecibo, where the machine spent all of Winter '92. At that time the FPTM only had 256 functional lags and some systematic noise that we did not understand. We concentrated on ISS observations, taking advantage primarily of the large bandwidth, and polarization observations of

millisecond pulsars, using here both the large bandwidth and the fine time resolution.

Since November '93 the FPTM has been at Parkes Observatory in Australia, undertaking a long term pulsar timing program. Most of the hardware bugs have been ironed out and the current effort is going into the observing and data analysis software. We routinely observe the bright millisecond binary pulsar J0437−4715, which has helped us immensely in the process of debugging the timing offsets.

Caltech graduate student Jagmit Sandhu is also working on the FPTM, his thesis project being the addition of another IF subsystem and digitizer, which will provide the capability to do dual frequency, dual polarization observations. We hope that the expanded machine will overcome the problems that potentially plague analog timing systems, and routinely do 100 ns timing or better.

2.3 The Data Path

It is useful to look at the functional block diagram of the FPTM in figure 2.1. It shows the major building blocks of the timing machine and the interconnections between them. This section focuses on the data path, from the telescope IF to the pulse arrival times.

The IF subsystem takes two IF radio signals from the telescope and downconverts them to baseband. The two IF signals are usually the two polarizations from the receiver and come at the same frequency between 0 and 1 GHz. This frequency may be the observing frequency itself, or some lower, intermediate frequency that the observing frequency is downconverted to for convenience. As an example, at Arecibo most receivers produce outputs at 260 MHz, and the control room has entire racks of equipment to process signals at 260 MHz.

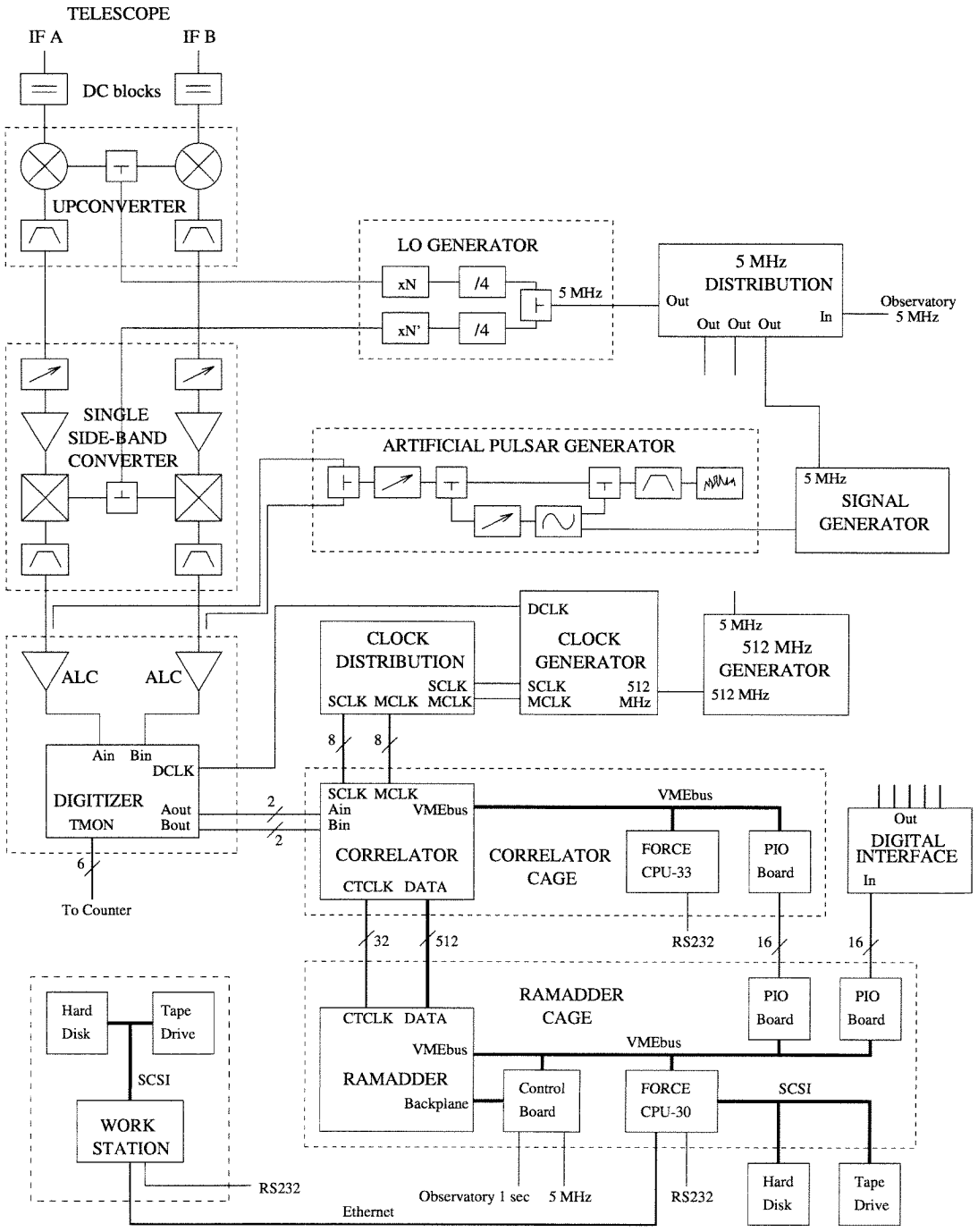
Inside the FPTM, the process of downconversion is done in three steps: upconversion to 1580 MHz, filtering through a 200 MHz bandpass filter, and single sideband downconversion to baseband or video frequencies. The initial upconversion and filtering allows us to use the baseband converter units from the Owens Valley Millimeter Array, which work with inputs in the 1–2 GHz range. The twin baseband converters also filter the baseband signals to 2, 8, 32 or 128 MHz of bandwidth, depending on the observing setup.

The digitizer takes the two baseband signals, that now have the correct bandwidth and power levels, and samples them at 256 MHz with a resolution of 2 bits. The sampling rate is independent of the bandwidth of the signals, even when it is less than 128 MHz. The digital outputs are ECL signals and they are buffered into several copies so that the digitized signals can be provided to several correlator boards at the same time.

The correlator boards can process signals from several sources, normally the digitizer or the output of other correlator boards. The source of input data for each correlator board is chosen from the software, and determines how the correlator will be configured. The two

Figure 2.1: A Block Diagram of the FPTM

The main functional blocks of the FPTM and the main interconnections between them. The signal path starts with the two IF signals from the telescope, which are upconverted and filtered, downconverted and filtered again, digitized, correlated, folded and read into the workstation. Each downconversion is done in single sideband mode (USB) with a 90° hybrid and a quadrature mixer. In addition to the two IF signals, the FPTM needs a 1 second time reference and a 5 MHz frequency reference from the observatory.



standard setups are autocorrelation with 256 lags per polarization, and crosscorrelation with 128 lags for each of the two polarizations and each of the two cross-polarizations.

The 512 lag data outputs from the correlator boards, with the 32 signals that clock them, go to the ramadder boards in the pulsar backend by means of a bundle of ribbon cable. In the ramadder boards each lag gets accumulated further and binned or folded into 1024 phase bins according to the rotational phase of the pulsar. Every minute or so the ramadder boards are read into the pulsar backend computer and stored onto a hard disk. The data set consists of 1024 lag profiles, an array of 512×1024 words. Thanks to a double buffering scheme, the ramadder boards can continue to accumulate data from the correlator while being read, so that no astronomical data is lost.

A raw dump of the ramadder boards is not very useful in itself, without being processed into pulse profiles. Once the ramadder boards have been read into the VMEbus computer, the data is transferred over the ethernet to a workstation and analyzed, summarized and archived. Two versions of the data are typically kept after the observations are over. The raw data from the FPTM, in the standard *correlator folded data* format, is written to tape, while the dedispersed pulse profiles are stored on the disk and used to obtain pulse times-of-arrival. The lists of arrival times are important because they reveal right away whether the timing system is producing results that are consistent with the previous observations, whether we can continue the observations or must debug the system immediately. In the long term, of course, the arrival times are the rotational history of the pulsar.

2.4 The Artificial Pulsar

The artificial pulsar generator simulates the signal from a pulsar at baseband. It is a broadband noise source modulated by an external signal generator. Pulsars of any period, duty cycle and strength can be generated. When the digitizer is connected to the artificial pulsar box instead of the baseband converter, the astronomical signal path bypasses the IF subsystem completely. Tests with the artificial pulsar isolate the digital part of the FPTM.

The observing routine requires that we always look at the artificial pulsar before the start of an observing run, as a way of testing the FPTM and ensuring that everything works properly by the time the observations begin. If we can see the pulsar at the the right period and in all the lags, then the control board in the pulsar backend must be providing the correct timing signals. A more subtle point is that the artificial pulsar must appear with the correct strength or signal-to-noise ratio, or else there must be an extraneous source of noise that is unaccounted for. The noise test is fundamental because during astronomical observations it is difficult to tell whether the noise levels are reasonable or not. After all, pulsars are intrinsically variable in many time scales and it would be easy to blame the lack of signal on the pulsar.

2.5 Timing and Support Signals

There are two fundamental signals that do not come from the telescope receiver but which are also external to the FPTM, and without which pulsar timing cannot be done. They are the 1 second UT reference and the 5 MHz frequency reference. The arrival times of pulsars are measured with respect to this globally accepted time, and the frequency reference ensures that we can exactly reproduce the pulsar frequency within the duration of an observation. At Parkes both references are generated from the same hydrogen maser (stability of $1{:}10^{14}$), and the resultant 1 second clock is regularly compared to a GPS signal obtained from the Deep Space Network station at Tidbinbilla. Typical errors are of less than 30 ns (see also §3.4).

All the clocks in the FPTM are tied to the 5 MHz reference. For this reason the 5 MHz reference goes through a distribution box that provides copies to several modules: the signal generator for the artificial pulsar, the LO generator, the clock generator for the digitizer and the correlator, and the control board in the pulsar backend.

The LO frequencies for the FPTM are generated from the 5 MHz reference essentially by frequency multiplication. The reference is first divided by 4 into 1.25 MHz, then multiplied by an integral factor up to the desired frequency between 900 and 1600 MHz. For this reason the possible LO frequencies are quantized and must be multiples of 1.25 MHz (see §5.2.2). The first LO is used to upconvert the telescope IF signals to the 1.5 GHz range, the second LO to mix them from 1.5 GHz down to baseband. The values chosen for the LOs depend on the IF band presented to the FPTM, which in turn depends on the receiver used. The most common observing and intermediate frequencies used at Parkes Observatory are listed in table 5.1.

There are over one thousand digital signals running around in the FPTM, some of them at frequencies as high as 256 MHz. The most important ones in the block diagram are the fast ones, namely the digitizer clock (DCLK) and the shift and multiply clocks for the correlator (SCLK and MCLK). DCLK goes to the digitizer module directly from the clock generator, while SCLK and MCLK are buffered and distributed to all eight correlator boards. The precise timing relationships between the three clocks need to be held constant at the ideal values, or else propagation delay errors would result in an incorrect correlation. The clocks are generated from a high stability 512 MHz clock oscillator that is phase locked to the observatory 5 MHz. The 512 MHz generator is very stable and will not lock to a reference that is more than 0.01% off from the nominal 5 MHz.

The 512 lags from the correlator are clocked into the ramadder boards by 16 clocks at a rate of 32 MHz. These clocks are generated independently by frequency division of MCLK, and will have random relative phases unless they are synchronized. Given that the 512 lags are treated synchronously in the ramadder boards, it is very important to phase lock the 32 MHz clocks not only with each other but with the ramadder 32 MHz clock

as well. This is described in detail in §8.5, as the advantages of a digital correlator and backend disappear if the clocks are out-of-phase and result in misscounts.

Chapter 3

The Millisecond Binary Pulsar J0437−4715

3.1 Introduction

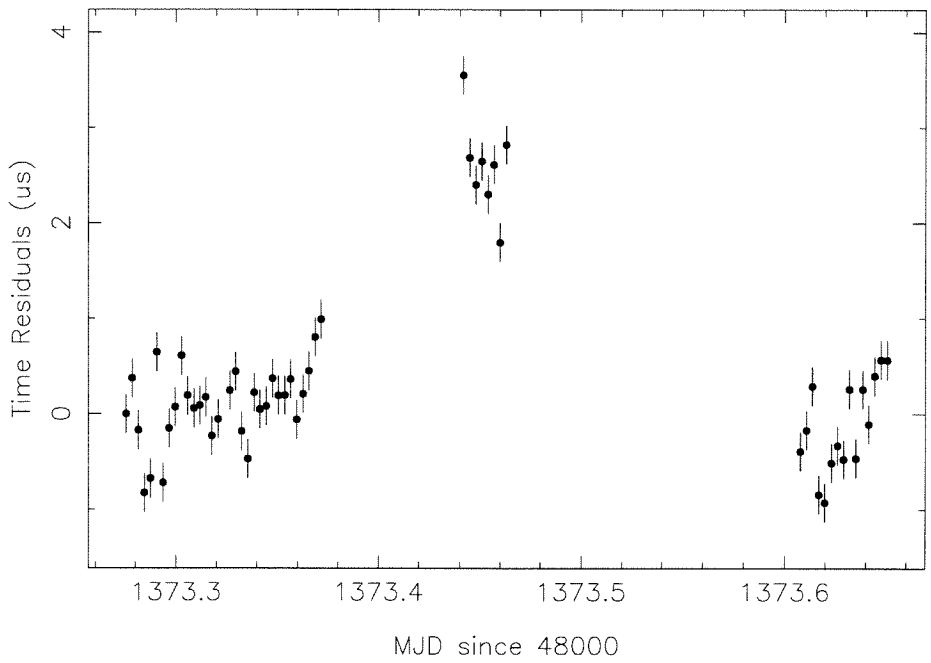
The Fast Pulsar Timing Machine is currently at Parkes Observatory, in Australia, where we have started a long term timing program of the southern pulsars. As part of the installation of the FPTM, we have been observing the bright millisecond binary pulsar J0437−4715. The observations have been performed in parallel with the filterbank system used for timing by the Australian team (Johnston *et al.* 1993), and have been extremely useful in the debugging of the FPTM. At the same time, they have helped us understand the limitations of both the FPTM and the Australian filterbank, and have already yielded some interesting results on the pulsar itself, which we present here.

PSR J0437−4715 was found in 1993 by Johnston *et al.* in a systematic survey of the southern sky with the Parkes radiotelescope. At a distance of 150 pc, J0437−4715 is the second closest known pulsar (after J0108−1431, see Tauris *et al.* 1994) and for that reason is also extremely bright, at times exceeding 1 Jy at 430 MHz. It is in a 5.74 day binary orbit around a 0.14 $M_\odot$ star, and has a spin period of 5.757 ms.

Its brightness makes it a useful calibrator for two reasons: first, because an accurate time-of-arrival (TOA) can be achieved in just a few minutes, and second, because while other pulsars fade away at high radio frequencies, J0437−4715 is still visible at frequencies of several GHz. This has allowed us to study the pulse profile and its many components in detail and as a function of frequency.

PSR J0437−4715 has been timed at Parkes since its discovery more than two years ago. There is a timing solution from the filterbank system that gives timing residuals of the order of a few μs. With the FPTM we are attempting to lower the residuals and improve the pulsar model and, at the same time, understand what the limitations are for timing

Figure 3.1: Single Night Timing Residuals With the FPTM
Pulse time-of-arrival residuals for J0437−4715 during a period of 9 hours, on 21-Jan-94. The observations were done at a central frequency of 1520 MHz with the 210 ft radiotelescope at Parkes Observatory. The arrival times were calculated from the pulse profile of the sum of the two linear polarizations. The residuals are deviations of the arrival times from those predicted by the pulsar timing model, and show an offset in the second group of points. This offset may be instrumental or due to an inaccurate pulsar model.



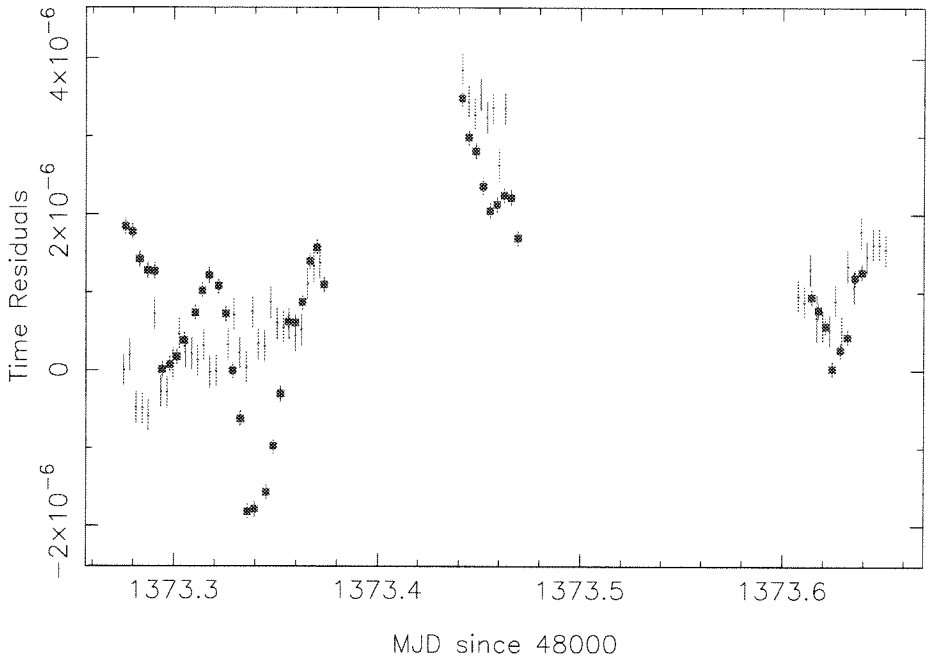
this and other pulsars.

3.2 Current Status of the Timing Results

Typical observing sessions last a few days and allow us to observe J0437−4715 for several hours per transit. Short integrations (5 minutes is typical) yield times-of-arrival accurate to within a fraction of a μs, depending on the observing frequency. The residuals from different observations are consistent as long as the observations are done within the same hour, but show drifts and jumps of as much as a few μs over several hours or days.

The drifts we have observed in the TOAs have been as large as 3 μs per hour, and the jumps could well be the result of a steady drift in TOA while we are not observing. Figure 3.1 shows an example of a jump of 2.5 μs that occurred within a single transit. Interestingly, the same drifts and jumps are seen by both the FPTM and the filterbank, which points that the causes of these timing errors are external to either timing system yet common to both through some shared signal (1 s or 5 MHz references) or resource (pulsar timing parameters). The TOAs from the two systems track each other faithfully,

Figure 3.2: A Comparison of Residuals From the FPTM and the Filterbank Pulse time-of-arrival residuals for J0437−4715 during a period of 9 hours, on 21-Jan-94, as deduced from both the FPTM and the Parkes filterbank. The FPTM data is the same as in figure 3.1 and is shown here as small circles (·) with dotted error bars. The filterbank residuals are pictured with solid squares (▪) and display drifts and wanders larger than those of the FPTM. The error bars in the filterbank data are underestimated and should be approximately ±0.5 μs.



the filterbank showing a bit more scatter than the FPTM (see figure 3.2).

The drifts and jumps seen in the PSR J0437−4715 data are never larger than a few μs, and in the long term the observed TOAs agree with the predicted TOAs to within several μs as well.

3.3 A Short History of the Residuals

We installed the FPTM at Parkes in November 1993, and from the very beginning the TOAs we obtained for PSR J0437−4715 produced residuals consistent with the Australian timing solution. By "consistent" I mean that the residuals were small compared with the pulsar period, in this case less than 100 μs compared with 5.75 ms. These residuals were still huge, though, compared with what we hoped to get, and we did not discover their origin until several weeks later, when we found that they followed a ∼6 day periodicity. This is significant because J0437−4715 has an orbital period of 5.7 days. It turned out that we were using the wrong directory for the pulsar ephemeris file, and this directory just happened to contain an old ephemeris file for J0437−4715, which predicted a slightly incorrect orbital phase.

This was exciting because the FPTM data was already producing TOAs good enough to start the feedback loop between pulsar model and residuals that is the foundation of pulsar timing.

The incorrect ephemeris could explain small drifts in the TOAs that amounted to large oscillations in the residuals, with a periodicity of P_b, but could not explain why some points were way off from the rest, sometimes by as much as 40 μs. It was usually the first timing point of every session or a point that had something special about it: the last point before a storm, the last point of the night, or a point from data contaminated by interference. The answer to this puzzle was again the incorrect ephemeris, as follows: using the incorrect ephemeris results *only* in folding at the wrong period. That means that as the integration progresses, the pulses will arrive earlier or later than they were predicted to (by the incorrect model), and will therefore smear over several phase bins. This smearing results in an effective delay (positive or negative) that is proportional to the length of the integration. Those special timing points turned out to correspond to shorter (or sometimes longer) integrations than the rest, generally because the integrations had to be aborted.

Given both ephemeris files, the correct and the incorrect one, it was now possible to correct the arrival times by fitting the calculated smear to the observed profiles. This technique worked but left the residuals at a level of some 10 μs, for two reasons. First, because it was difficult to account completely for the smear produced by the wrong period (because the folding period is updated irregularly during the integration). And second, because some of the observations were at different frequencies (1.4 and 2.4 GHz) and the effective DM for the FPTM is different than the DM obtained with the filterbank.

Dispersion measure is a well defined number for each pulsar, but in practice is very difficult to measure accurately because it requires the comparison of arrival times at two different frequencies. At the two frequencies the pulse profiles can be sufficiently different that it is impossible to assign fiducial points to the profiles that correspond to identical rotational phases (Phillips 1991). Even for simple profiles with a single pulse, a slight change in the pulse shape with frequency brings uncertainty because it is not clear whether the pulse peak should be used (as R. Manchester does), or the pulse center from a weighed average, or the phase of the first harmonic (as J. Taylor uses). Hence each timing system must use its own, yet consistent, dispersion measure.

To complicate matters even more, different timing systems produce different profiles, so that even using the same method to calculate the dispersion measure won't help. This was particularly important for us because the FPTM resolves the main pulse for J0437−4715 and the filterbank system doesn't. In short, we were using the wrong DM for this pulsar as observed through the FPTM, and it resulted in large timing residuals. Unfortunately, we didn't yet have enough timing data to calculate what the correct DM should be for the FPTM.

At this point we started observations with the correct ephemeris, and soon discovered another problem, that the observatory position was off by some 300 meters in the timing files. This corresponds to peak-to-peak daily residuals of the order of a μs. Correcting the ephemeris file didn't make the residuals go to zero, unlike we had hoped. Or even 1 μs, as we expected. The current status of the residuals is discussed below.

3.4 Understanding the Current Residuals

Often a group of observations of J0437−4715 done within the same hour will yield residuals that cluster tightly, the scatter being fully consistent with the timing errors due to the finite signal-to-noise ratio of short integrations (the radiometric limit). Since observational errors are of the order of 0.2 μs at 1500 MHz (and up to a few μs at other frequencies), the goal of this timing program is to generate a model that can predict the observed TOAs to within a fraction of that, namely a few ns, over years of observations. When the observed residuals are larger than the observational error (as they are in figure 3.1), there has to be an error either in the observed TOA or in the pulsar model. In this section we discuss the possible causes of these large residuals, be they due to the pulsar, the timing machine or the pulsar model.

One of the prime candidates for the existence of drifts with a time scale of a few hours was that our observations were in linear polarization, not circular polarization (we have since changed to circular). Pulsars can have a large fraction of linear polarization, and as a result can produce very different profiles in the two linear polarizations. The two profiles will change with parallactic angle as the telescope feed rotates and the projection axis changes. With a perfect receiver there would be no problem in adding the two profiles to obtain a total intensity profile, but differential gains in the two IF channels can upset this method. The total intensity I can be written as

$$I = g_1 x + g_2 y = g_1(x + y) + (g_2 - g_1)y, \tag{3.1}$$

where x and y are the detected signals in each receiver channel and g_1 and g_2 are the two gains. The last term in equation (3.1) changes with time and has a relative magnitude of $\Delta g/g$. Pulsars have little circular polarization and the differential gain will not change the shape of the intensity profile, hence the switch in the receiver feed to circular polarization.

In March 1993 a hybrid was installed at Parkes and we tried observing with circular polarizations for the first time. That did not eliminate the drifts we had observed earlier, which must therefore have a different cause. Nevertheless observing with circular polarization is important and we will continue to use the hybrid.

Another strong candidate for the excess residuals is the Parkes Observatory clock, the time standard that both the FPTM and the filterbank use as reference. Any errors in this

reference signal would be observed by both systems as an error in the pulse time-of-arrivals, which would be consistent with what we observe (see figure 3.2). However, drifts of a few microseconds are enormous and would have been already detected in VLBI experiments. We are assured by VLBI observers (John Reynolds, pers. comm.) that the observed drifts are of 70 ns per day and that peak-to-peak residuals are of no more than 20 ns. Therefore the clocks appear to be reliable.

It is of course possible that our pulsar model needs to be adjusted so that it accounts for the drifts. Currently the model is good in that we never measure residuals larger than $\sim$10 μs. However, the residuals need to be randomly scattered about zero (the model prediction) and they are not. We have only recently started to get the higher quality timing data, and we are hoping that with a few more timing runs we will obtain a better fit that accounts for the residuals. In particular, the existence of the drifts ruins the parallax fit which the Australian team had, and we need more timing points to get an improved timing solution.

Finally, the last thing that is shared by the two timing systems is the pulsar signal itself. Despite having very different hardware, both systems obtain the TOAs in similar manners, which are therefore suspect. We are currently learning the idiosyncrasies of the radio data and how to become more immune to errors such as bad channels, radio interference and scintillation islands and gradients. In addition, we are experimenting with different timing algorithms: how to fit templates to profiles, whether fitting to individual channels instead of the dedispersed profile is feasible, what improvement we get from calculating the TOA at the middle of the integration instead of at the beginning, and in general just getting a better handle on how to do submicrosecond timing. The problem is difficult because we are trying to push pulsar timing to consistent submicrosecond residuals.

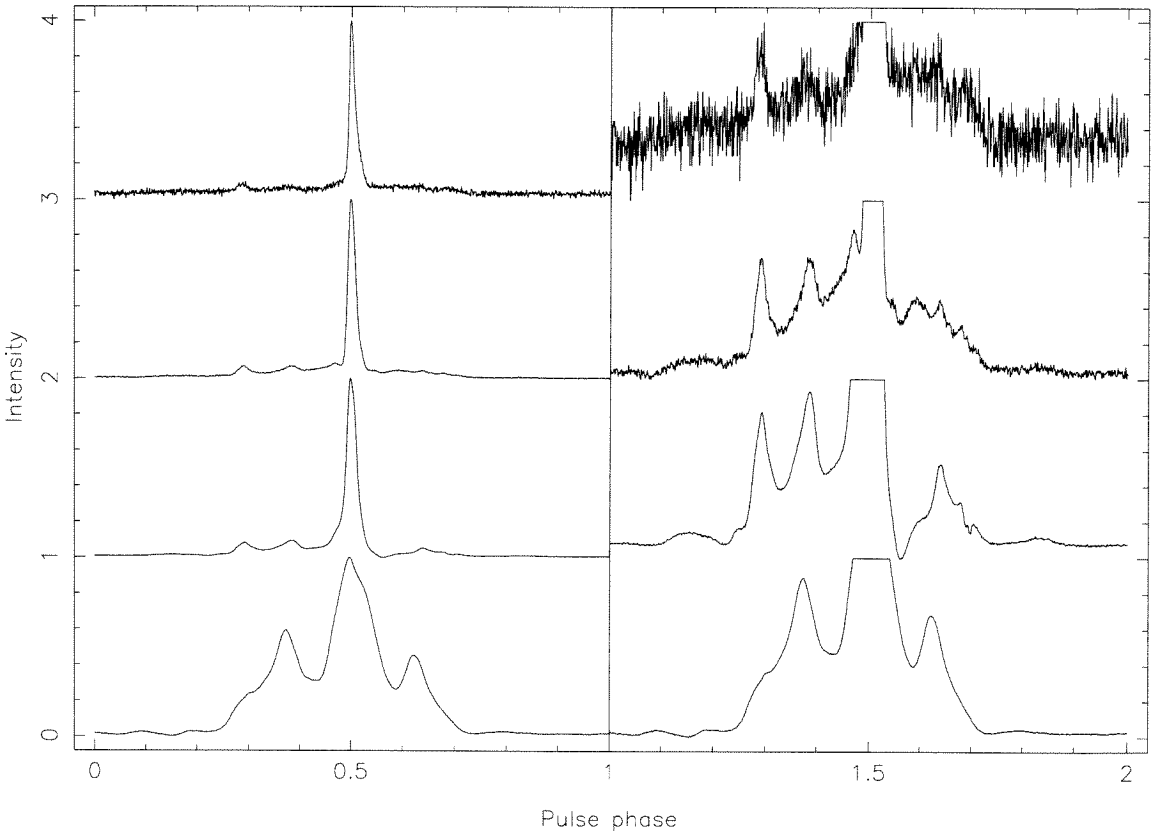
3.5 Discussion

The pulse profile of J0437−4715, shown in figure 3.3, is striking because it clearly presents as many as 10 components besides the main pulse, while most pulsars have at most a total of 5 components in their profiles (see Rankin, Stinebring, and Weisberg 1989 and related papers). These components are distributed along the entire pulse period and have a peculiar frequency dependence. Polarization profiles will be extremely useful in order to understand the alignment of the pulsar spin with the binary orbit, but we don't yet have polarization profiles because we have been waiting for the installation of the circular polarization setup.

It is not clear why J0437−4715 should be so different from all the other pulsars whose profiles have been studied in detail, but we must note a few points: first, that J0437−4715 is extremely bright and therefore very high signal-to-noise ratio profiles can be made. Since

Figure 3.3: The Pulse Profile of J0437−4715

Integrated pulse profiles of J0437−4715 at the observing frequencies of 436, 1520, 2400 and 4680 MHz (from bottom to top). In the left-hand panels the intensity is normalized to a peak of unity height. The right-hand panels are the same profiles but multiplied by 10 in intensity (except the lowest panel, which is multiplied by only 1.5). Notes: (*a*) the small dip after the main pulse in the 1520 MHz profile is due to instrumental effects. It can also be seen in the 2400 MHz profile. (*b*) the baselines have been chosen somewhat arbitrarily and do not represent the actual intensity, even though there is clear emission throughout most of the period. (*c*) The profiles change immensely with frequency, and the main pulse can be resolved into two pulses when going from 1520 to 2400 MHz. The full widths of the main pulse at half max are, from bottom to top: 533, 134, 106 and 81 μs. Note that a feature 81 μs wide corresponds to an observed emission region of 24 km.



some of the components are much weaker than the main pulse (less than 1% in intensity), this fact alone might explain why weak components might be seen here but missed in other pulsars.

Second, it is possible that millisecond pulsars have intrinsically different profiles from ordinary pulsars. We could be seeing emission from much closer to the neutron star's surface and evidence for higher magnetic field moments. This would be an exciting result because it would challenge the way we estimate pulsar magnetic fields and spin-down ages, and therefore millisecond pulsar evolution. PSR B1534+12 (Wolszczan 1991) also presents emission over much of its profile, although its components are no more than five, and the one other pulsar that could show this effect, B1937+21, shows no features other than the main pulse and a bright interpulse.

We are studying PSR J0437−4715 from two points of view. First, that of precision timing. J0437−4715 is an ideal pulsar to conduct timing experiments on. So far we have not reached the microsecond level due to instrumental problems, but we should able to go beyond that in the next few timing runs if progress continues as it has until now. At the same time, we are studying the pulse profile of J0437−4715 and its components as a function of frequency. In the next few months we will observe with cross-polarization data and obtain pulse profiles in all four Stokes parameters. We should be able to measure the position angle swings through the entire profile, which will give us a good handle on the emission geometry of this strange pulsar with so many components.

Chapter 4

Discovery of a Luminous Binary Millisecond Pulsar

Abstract

We report the discovery of a field binary millisecond pulsar, J0218+4232, with a period of 2.3 ms and in a 2.0 day binary orbit around a 0.16 $M_\odot$ companion. The new pulsar was serendipitously discovered as a steep-spectrum, highly polarized, compact radio source during imaging observations at WSRT, and was later confirmed to be a pulsar with observations done at Jodrell Bank. With a dispersion measure of 61 pc cm^{-3}, it lies well outside the electron layer in the direction (l=140°, b=-18°). At a distance of more than 5.7 kpc, it is the farthest known field millisecond pulsar and has a radio luminosity L_{400} comparable to that of PSR B1937+21.

4.1 Introduction

Systematic, untargeted searches for millisecond pulsars have resulted in the discovery of a dozen nearby recycled pulsars in the past few years (Camilo 1994). It is now clear that the Galaxy contains some 10^5 millisecond pulsars (Lorimer 1994), of which these searches probe primarily the local kpc.

In contrast with the untargeted searches, PSR J0218+4232 was discovered as a steep spectrum compact source in unrelated imaging observations at the Westerbork Synthesis Radio Telescope (WSRT). Follow-up work revealed that it was highly polarized, and thus a prime candidate for a millisecond pulsar. The actual detection of pulsations was difficult because of the short period, large dispersion measure and steep spectrum, which made the pulsar extremely dispersed at 325 MHz and rather weak at 1.4 GHz. Despite its brightness at the lower frequencies, it is likely that it would not have been detected by any of the ongoing blind searches, and therefore belongs to pulsar parameter space that is not actively being searched.

Intrinsically luminous pulsars like B1937+21 are expected to be rare (Johnston and Bailes 1991) and will be difficult to find by chance. PSR B1937+21 was detected in a large

survey for scintars (Backer *et al.* 1982), while J0218+4232 was discovered serendipitously. New pulsars with large DM/P like these two will be hard to detect and are therefore prized discoveries.

4.2 Observations and Analysis

In a program to study the low frequency spectrum and variability of the radio supernova SN1986j (Rupen *et al.* 1987) in the spiral galaxy NGC891, one of us (A.G. de Bruyn, in collaboration with J.M. van der Hulst) had acquired a long series of WSRT observations at frequencies of 325 and 610 MHz. The large field of view synthesized at 325 MHz (2.7 degrees at halfpower) results in the detection of numerous background sources, those at the edge of the field appearing to be polarized due to instrumental effects. When testing software to correct for this off-axis polarization, we discovered that a relatively weak source, about 50' west of the galaxy NGC891, had an anomalously high linear polarization which also changed during the 12 hour synthesis. This meant that the source had to be intrinsically polarized, with an average 40% linear polarization, much higher than the expected instrumental polarization of 5% at that location.

Inspection of the 610 MHz observations of the same field revealed the source to have an extremely steep spectrum ($\alpha = -3$, where the radio flux $S_\nu \propto \nu^{-\alpha}$). Observations from 1986 to 1991 also revealed it to be highly variable from one synthesis to the next. These properties are reminiscent of PSR B1937+21 and made the new source a prime candidate for a millisecond pulsar.

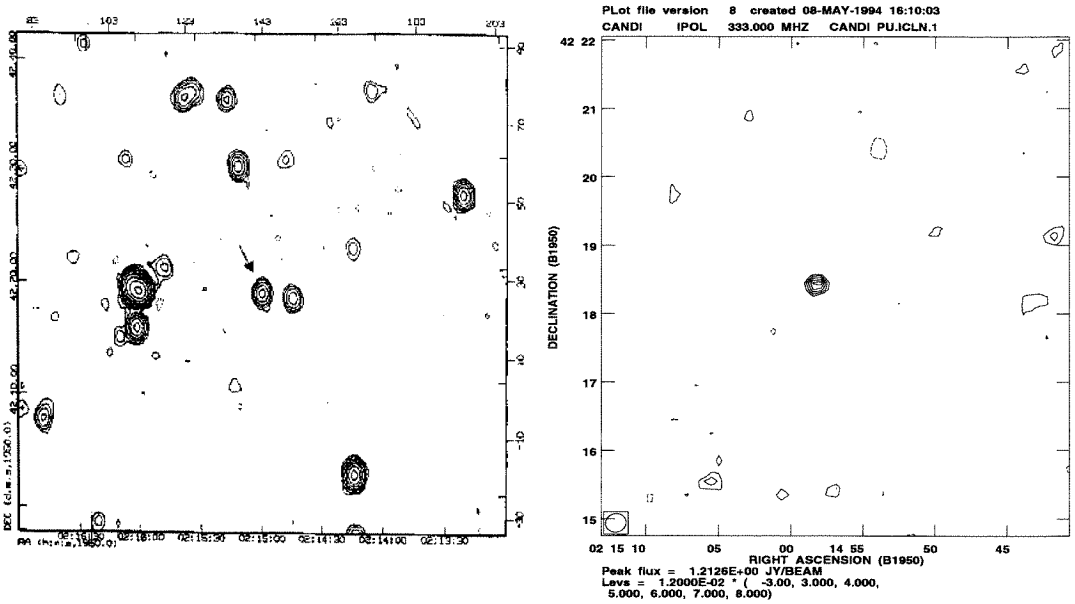
From the WSRT observations we calculated a rotation measure of -61 rad m^{-2}, similar to that of the two nearby extragalactic sources 3C66B and 3C65 at offsets of 1.1 and 2.6 degrees and with rotation measures of -67 and -81 rad m^{-2} respectively (Simard-Normandin, Kronberg, and Button 1981). The conclusion was that our source might be behind most of the Galactic Faraday layer in that direction, which led us to expect a large dispersion measure, $\geq$50 pc cm^{-3}.

In January of 1992 we observed the candidate source at the Very Large Array (VLA) at 333 MHz and at 1.4 GHz (see figure 4.1). We observed again in February 1992 in phased array mode and searched for pulsations with the High Time Resolution Processor (HTRP, see McKinnon 1992), but we were not able to find any signals significantly above the noise. At 333 MHz the HTRP sampled a 1×4×0.125 MHz (single polarization, 4 channel) filterbank every 0.3 ms, while at 1.4 GHz we tried both a 2×14×2 MHz filterbank sampled every 1.24 ms and a 2×14×4 MHz filterbank sampled every 0.67 ms. No pulsations were discovered.

At that time it became clear that if the candidate was indeed a millisecond pulsar, it had to both be a fast one and have a large dispersion measure, which made continuing

Figure 4.1: P-band Images of PSR J0218+4232

Two synthesis maps of J0218+4232. The left image is from WSRT at 325 MHz and the right image is from the VLA in A-array, at 333 MHz. In the WSRT image an arrow points to J0218+4232, which is also the central source in the VLA panel. In both cases the beam sizes are comparable to the apparent size of the candidate source. The fields of view are 45' and 7' respectively.



the searches all the more interesting. An apparent variable rotation measure also led us to consider a binary model in which a millisecond pulsar might be ablating the companion, *à la* PSR B1957+20 (Fruchter, Stinebring, and Taylor 1988), or perhaps precessing. In order to beat both the large assumed dispersion smearing and the known steep spectrum, we tried to record the baseband signal at low frequency with VLBI MkIII recorders (Clark and Rogers 1982; Wrobel 1991), then play it back onto a workstation and do coherent dedispersion of the baseband signal in software. For this we designed and built a special-purpose digital buffer board and put it in a VMEbus crate with a computer, a large hard disk and an Exabyte tape drive (see appendix D). We then interfaced this VMEbus system with a MkIII recorder in playback mode, and recorded one of the 4 Mbit/s data streams from the recorder through the buffer board and onto the hard drive, and later onto tape. Once the data was on an Exabyte tape, it could be analyzed on a Sun workstation.

In March and April 1992 we recorded the phased array baseband signal of this source and PSR B1937+21 at frequencies of 325 and 610 MHz, with the VLBI MkIII recorders at WSRT. Shortly afterwards, in May 1992, we recorded data in the same manner at the VLA at 327 MHz. Unfortunately, the VLA MkIII tapes were corrupted because there were head positioning problems in the recorder during our observing session and each pass through the tapes overwrote the previous pass. Despite having the WSRT tapes, we gave up this

approach due to lack of manpower for the data analysis.

A literature search showed that the source was also in the 6C catalog (Hales, Baldwin, and Warner 1993) with a flux of 0.66 Jy at 151 MHz, and in a 34.5 MHz survey (Dwarakanath and Shankar 1990) with a flux of 30–50 Jy. These independent observations confirmed the spectral index of -3, one of the largest among pulsars.

The position of the source was determined to an accuracy of about 1".5, sufficient to search for an optical identification. None was found to a limit of 20m on the POSS-I plates. CCD photometry on the Nordic Optical Telescope in January 1993 revealed no optical sources down to 22.5m in R.

The next attempt at detecting pulsations from the candidate was made in February 1993 at Jodrell Bank, where we recorded data at 411 MHz in two configurations: a $2\times32\times31.25$ kHz filterbank sampled every 150 μs, and a $2\times64\times125$ kHz filterbank sampled every 300 μs, in both cases with one bit of resolution per channel. We used the Caltech pulsar search software (Deich 1994) for the data analysis, which consisted in dedispersion at a number of trial dispersion measures, and then a Fourier search for periodicities in each of the dedispersed data streams. It was in this data where we finally detected pulsations at a period of 2.3 ms in all on-source files and in none of the off-source files. The dispersion measure for this signal was 61.2 pc cm^{-3}, confirming our expectations. The 2.3 ms periodicity was confirmed in April 1993 by the same method.

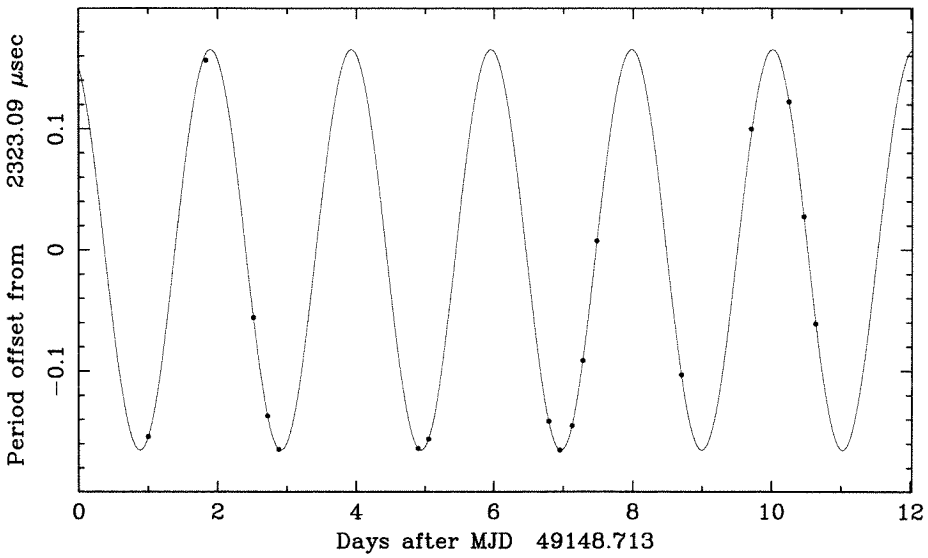
Subsequent observations at Jodrell Bank have revealed that the pulsar is in an almost circular, 2.03 day binary orbit (see figure 4.2). The observations of PSR J0218+4232 now span more than a year and give the timing solution shown in table 4.1. No radio eclipses are seen, although in some epochs we have been unable to detect the pulsar as a radio source at 1.4 GHz with the VLA, with a rms noise of only 0.25 mJy.

4.3 Discussion

The dispersion measure of PSR J0218+4232 places it outside the electron layer adopted in the model of Taylor & Cordes (1993). According to this model, the maximum dispersion measure in the direction $l = 140°, b = -18°$ is only 52 pc cm^{-3}, and as a result the pulsar must be further than 5.7 kpc away and at least 1.75 kpc below the plane of the Galaxy. In this direction the free electron density model so very poorly determined that we cannot really say how far the pulsar is.

However, it is clearly extremely bright and only because of this could it be detected. Due to selection biases it is premature to conclude that its detection lends support for a large scale height for field millisecond pulsars, and we can only assert that it is consistent with it. An minimum initial velocity of $\sim$80 km s^{-1} in the z direction would be needed to send this system out to $|z| > 1.75$ kpc at a Galactocentric radius of 12 kpc, had it been born

Figure 4.2: Orbital Period Fit for J0218+4232
The apparent barycentric spin period of J0218+4232 is modulated by the binary period. A sinusoidal fit to the apparent periods is shown, revealing the 2.029 day periodicity. The eccentricity of the orbit is very small and is limited to <0.00002 by the current observations.



in the plane itself (Hartmann, Epstein, and Woosley 1990, and Paul Ray, pers. comm.).

The measured period derivative for J0218+4232 is $0.97\pm0.41\times10^{-20}$, one of the smallest known. It gives the pulsar a characteristic age of 3.8×10^{9} yr, a magnetic field of 1.5×10^{8} G and a spin-down luminosity of 3×10^{34} erg. PSR J0218+4232 lies in the lower left of the B–P diagram, among the millisecond pulsars with lowest magnetic fields.

The orbital parameters quoted in table 4.1 yield the mass function

$$f_1(m_1, m_2, \sin i) = \frac{(m_2 \sin i)^3}{(m_1 + m_2)^2} = \frac{4\pi^2}{G}\frac{(a_1 \sin i)^3}{P_b^2} = 0.00204 M_\odot, \qquad (4.1)$$

where m_1 and m_2 are the pulsar and companion masses, P_b is the binary period, and $a_1 \sin i$ is the orbital semi-major axis projected onto the line-of-sight. For a typical pulsar mass m_1=1.4 $M_\odot$, the companion mass would be $m_2 \geq 0.16\ M_\odot$, very similar to those of other millisecond binary pulsars (see table 4.2).

We have an upper limit of 0.00002 to the orbital eccentricity. This upper limit is consistent with the models of Phinney 1992, which predict an eccentricity at least an order of magnitude smaller. It will be hard to measure ϵ down to the level of 10^{-6}.

Preliminary analysis of the timing data gives timing residuals of the order to 70 μs. A proper analysis combined with data taken over a longer time span should make J0218+4232 an interesting source for the timing array (Foster 1990), especially because of its location on the sky away from other millisecond pulsars.

It is somewhat surprising to see that fast, bright radio pulsars still remain to be dis-

covered in the Northern skies. In the case of PSR J0218+4232 this is because few searches have been conducted that are sensitive to the combination of millisecond periods and large dispersion measures. Even the current Arecibo searches would have missed this pulsar because the dispersion smearing across each 250 kHz filterbank channel at 430 MHz is more than half a period. It is unfortunate that this region of parameter space is undersampled, as discovering a pulsar faster than B1937+21 would have important implications toward the equation of state of dense matter (see Kulkarni 1992 and references therein).

Table 4.1: Parameters of PSR J0218+4232

Parameter	Value
Interferometric Solution:	
Right ascension (B1950)	$02^h14^m58^s.49$ ($0^s.15$)
Declination (B1950)	$+42°18'27''.0$ ($2''.0$)
Flux density at 34.5 MHz (Jy)	30–50
Flux density at 150 MHz (mJy)	660
Flux density at 325 MHz (mJy)	100–200
Flux density at 608 MHz (mJy)	26
Flux density at 1400 MHz (mJy)	1–2
Rotation measure (rad m^{-2})	-61
Timing Solution:	
Right ascension (J2000)	$02^h18^m06^s.381$ ($0^s.001$)
Declination (J2000)	$+42°32'16''.55$ ($0''.06$)
Epoch of period (MJD)	49150.6086
Period (s)	0.00232309045722 (4)
Period derivative (10^{-20})	0.97 (41)
Dispersion measure (pc cm^{-3})	61.2520 (9)
Orbital period (s)	175292.288 (5)
Projected semi-major axis (lt-s)	1.98444 (1)
Eccentricity	< 0.00002
Time of ascending node (MJD)	49150.608838 (4)
Mass function ($M_\odot$)	0.00203844 (4)
Derived Parameters:	
Spin-down age (yr)	3.8×10^9
Magnetic field strength (G)	1.5×10^8
Companion mass ($M_\odot$)	0.16
Galactic longitude (degrees)	139.58
Galactic latitude (degrees)	-17.72
Distance (kpc)	> 5.7
Spin-down luminosity (erg s^{-1})	3×10^{34}
Radio luminosity, L_{400} (mJy kpc^2)	> 2700
Spectral index	-3

Table 4.2: Parameters of Field Low-Mass Binary Pulsars

References: (1) Bailes *et al.* 1994, (2) Johnston *et al.* 1993, (3) Lorimer *et al.* 1994, (4) Wolszczan 1990, (5) Foster, Wolszczan, and Camilo 1993, (6) Segelstein *et al.* 1986, (7) Backer *et al.* 1982, (8) Boriakoff, Buccheri, and Fauci 1983, (9) Fruchter, Stinebring, and Taylor 1988, (10) Nice, Taylor, and Fruchter 1993, (11) Camilo, Nice, and Taylor 1993, (12) Camilo (pers. comm.).

Pulsar	P	$\dot{P}$	DM	P_b	e	$M_c \sin i$	D	S_{400}	L_{400}
	ms	10^{-20}	pc/cm^3	days	10^{-3}	$M_\odot$	kpc	mJy	mJy kpc^2
J0034−0534[1]	1.88	0.67	13.8	1.59	<0.1	0.14	1.0	16	16
J0218+4232	2.32	0.97	61.2	2.03	<0.02	0.16	>5.7	100	>3250
J0437−4715[2]	5.76	4.5	2.6	5.74	0.0182	0.14	0.15	600	13.5
J0613−0200[3]	3.06	1.1	38.8	1.20	<0.022	0.13	2.2	24	115
J1045−4509[1]	7.47	1.9	58.2	4.08	0.0191	0.16	3.2	20	200
B1257+12[4]	6.22	12.1	10.2				0.6	20	7.2
J1456−3327[3]	7.99	<0.6	13.6	76.2	0.167	0.27	0.7	17	8.3
J1643−1221[3]	4.62	3.3	62.4	147.0	0.506	0.13	>4.9	93	>2200
J1713+0747[5]	4.57	0.81	16.0	67.8	0.0741	0.28	0.8	36	23
J1730−2304[3]	8.12	<1.9	9.6				0.5	50	13
B1855+09[6]	5.36	1.78	13.3	12.32	0.0215	0.23	0.7	31	15
B1937+21[7]	1.56	10.7	71.0				3.6	240	3100
B1953+29[8]	6.13	2.97	104.6	117.35	0.331	0.17	5.4	15	440
B1957+20[9]	1.61	1.68	29.1	0.38	<0.04	0.02	1.5	20	45
J2019+2425[10]	3.93	0.83	17.2	76.5	0.112	0.32	0.9	2.7	2.2
J2145−0750[1]	16.05	<2.	9.0	6.84	0.0209	0.43	0.5	50	12.5
J2229+2643[12]	2.98	?	23.0	93.0	0.25	0.12	1.2	?	?
J2235+1506[11]	59.78	16.	18.1				1.2	3	4.3
J2317+1439[11]	3.45	0.47	21.9	2.46	0.0012	0.20	1.9	22	80
J2322+2057[10]	4.81	0.7	13.4				0.8	0.5	0.3

Chapter 5

The IF Rack

Abstract

The IF rack of the FPTM conditions the two astronomical radio signals from the telescope for the sake of the digitizer. This chapter describes the entire IF chain in detail, including the IF processing done at the telescope prior to the FPTM. A global description with block diagrams can be found in chapter 2.

5.1 Introduction

The Parkes radiotelescope is a 64 m alt–az paraboloid that can see as far north as +25° of declination. It has receivers for frequencies from 75 MHz up to 5 GHz, and can provide a useful radio signal to the FPTM in almost every configuration. Besides the two IFs from the telescope and in addition to AC power, the FPTM requires only two other signals from the observatory, namely a time reference for absolute timing and a frequency reference for stable operation. The absolute time reference is the VLBI 1 PPS signal and the stable frequency reference is the observatory maser, distributed in the form of a 5 MHz RF signal.

The telescope receiver provides two astronomical radio signals to the equipment in the control room. Both signals are at the same frequency and correspond to the two polarizations states, be they linear or circular. Observing frequencies above 1 GHz are downconverted in the receiver cage and observing frequencies below 1 GHz are left unconverted, so that all signals in the control room are in the range 0–1 GHz.

The primary purpose of the IF rack or subsystem is to convert the radio signals from the telescope receiver down to baseband. There are two identical channels in this subsystem, one per polarization. During the process of downconversion, the two IF signals are filtered and amplified so as to present the digitizer with clean baseband signals that have the correct power levels. The downconversion occurs in two stages, an initial upconversion to the 1.5 GHz range and a final downconversion to baseband, after which the signals are filtered to the final bandwidth. The rationale for this apparently quixotic method is that the wideband downconverters that we cloned from OVRO work in the 1–2 GHz range.

The FPTM is currently located in the control room of the Parkes telescope tower. It has a host computer that one can log onto and run programs on, but it can also be controlled remotely by programs running on a Sun workstation. This same workstation normally controls the telescope, and is ideally in charge of both the telescope and the FPTM during an observation. The data taken by the FPTM is immediately transferred to the workstation for archival to tape and for possible on-the-spot reduction.

5.2 Frequency Conversions

The intermediate frequency rack is the part of the FPTM that accepts radio frequency signals from the telescope, filters them and converts them down to baseband where they can be digitized. There are two identical IF channels, one for each polarization. Although in principle the two channels could be used to downconvert signals from different sky frequencies, at the moment they work in parallel and therefore at the same frequency.

5.2.1 At the Telescope

Before connecting the two IF signals from the telescope to the FPTM, it is important to know both their frequency range and power levels. The power levels are fundamental because if they are too large they can damage the FPTM, and if they are too small the FPTM will not get enough signal. The frequency of the IF signals, or rather the exact conversion from sky frequency to IF frequency, is necessary for the IF subsystem to position itself in the range of IF frequencies that corresponds to the desired range of sky frequencies.

Sky frequencies above 1 GHz are mixed at the telescope receiver down to the range 0–1 GHz (see table 5.1). The local oscillator for this downconversion ($f_{\rm LO}$) is generated by multiplying a ~100 MHz signal by an integral factor of 17–50. The ~100 MHz signal is generated in the control room with a Hewlett-Packard frequency synthesizer and is then sent to the telescope receiver cage, where it is multiplied by the given factor and mixed with the receiver output. The actual values of the ~100 MHz frequency and the multiplication factor depend on the desired sky frequency $f_{\rm SKY}$ and, once set, will typically remain unchanged until the next receiver change. It is important to know and verify these two numbers before observing: while the HP frequency is displayed in the control room on the front panel of the Hewlett-Packard generator, the multiplication factor is set in the receiver cage and can only be found in the receiver logs. Therefore, it is fundamental to generate a pure tone during the observing setup and verify that it comes down at the correct frequency in the IF.

Whether the telescope IF is the upper sideband or the lower sideband depends on the particular configuration. With USB conversion we have $f_{\rm IF} = f_{\rm SKY} - f_{\rm LO}$, while with LSB conversion we have $f_{\rm IF} = f_{\rm LO} - f_{\rm SKY}$.

Table 5.1: IF Configuration at Parkes

The most common observing configurations used for pulsar timing at Parkes Observatory. Since the actual sky and IF center frequencies ($f_{\rm SKY}, f_{\rm IF}$) can vary, one must make sure that the frequencies add up, from the (possible) initial downconversion in the receiver cage through the final upconversion and downconversion in the FPTM. In LSB downconversion increasing IF frequencies correspond to decreasing sky frequencies. Typical setups for the FPTM include the frequency/bandwidth pairs 436/32, 660/32, 1520/128, 2400/128 and 4800/128. All frequencies and bandwidths are in MHz.

SKY $f_{\rm SKY}$	BW	**Telescope LO** $f_{\rm LO}$	**IF** $f_{\rm IF}$	**Sideband**
436.00	32		436.00	USB
660.00	32		660.00	USB
1520.04	400	$107.78 \times 18 = 1940.04$	420.00	LSB
2400.00	200	$110.00 \times 19 = 2090.00$	310.00	USB
4780.00	320	$106.25 \times 48 = 5100.00$	320.00	LSB

5.2.2 The Upconverter

Upon entering the FPTM upconverter, the two IF signals are immediately upconverted to the 1.5 GHz region, where they go through a 200 MHz bandpass filter centered at 1580 MHz (see figure 5.1). The IF channels have a mixer and a filter each, but they share LO frequencies. This could be a problem if we wanted to obtain the two IFs from different regions of the observing band and the telescope instruments couldn't provide them at the same IF, but in general the two IF signals are the two polarizations of the same observing frequency.

The upconversion LO frequency $f_{\rm UP}$ is set from the software and is chosen according to the IF center frequency $f_{\rm IF}$ (which ultimately depends on the desired sky center frequency $f_{\rm SKY}$). Namely, the local oscillator in the upconverter converts $f_{\rm SKY}$ to 1580 MHz and therefore $f_{\rm UP} = 1580 - f_{\rm IF}$.

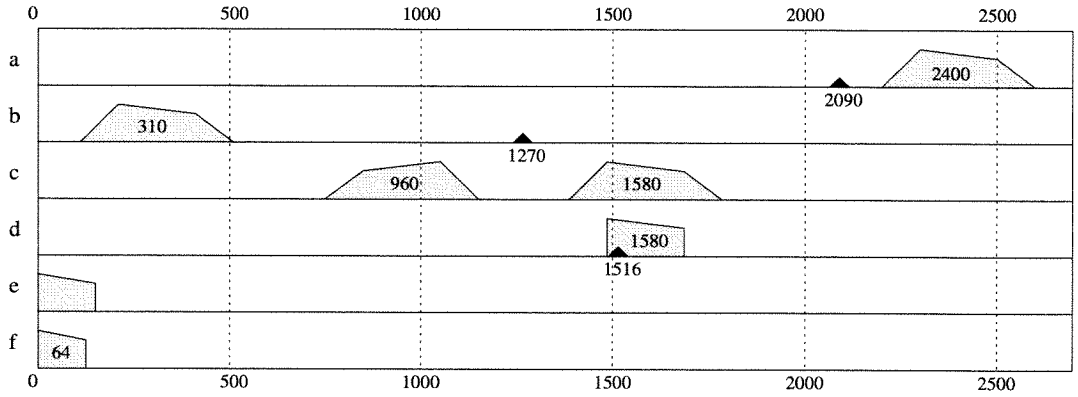
The LO generator board is based on a frequency synthesizer board (the Qualcom model Q0410) that has two limitations: it only generates frequencies between 900 and 1600 MHz, and they are quantized in steps of 1.25 MHz. As a result we are limited to IFs below 680 MHz, which is not a problem with the receivers we use at Parkes (see table 5.1). Two solutions are possible if it ever becomes necessary to process IFs above 680 MHz: to buy or borrow a frequency generator that can go below 900 MHz, or to buy filters centered at frequencies higher than 1580 MHz and frequency generators that can go above 1600 MHz (for downconversion, see below). The current mixers could still be used at the new frequencies.

The fact that the LO frequency is always a multiple of 1.25 MHz means that we can only select the observing frequency with an accuracy of 1.25 MHz. This is not generally a problem because for pulsar observations being able to choose an exact center frequency is

Figure 5.1: Downconversion of the 2.4 GHz Band

This figure shows the steps involved in the downconversion of the 2.4 GHz receiver output. The passbands are shown with a sloping top as a way to follow which sideband is which and are labeled with their central frequency. The LO frequencies are marked with triangles. All frequencies are in MHz.

The receiver output at 2.4 MHz is mixed with a 2090 MHz LO (a) to produce an IF at 310 MHz (b). This IF is upconverted to the 1580 MHz range (c), where it goes through a 200 MHz filter to eliminate the lower sideband (d). A 1516 MHz LO is then used for single sideband conversion to baseband (e) and finally the baseband signal is filtered through a 128 MHz filter (f).



not important, as long as we know exactly what frequency we end up observing at. The only inconvenience is that sky frequency calculations become more complicated (§5.2.4 gives a few examples). Incidentally, the LO generator used for downconversion is also a Q0410 and suffers from the same problems.

The 200 MHz filter eliminates any LO signal $f_{\rm UP}$ that filters through the mixer as well as the lower sideband (at $f_{\rm LSB} = 1580 - 2f_{\rm IF}$ MHz) but lets the upconverted signal at $f_{\rm USB}$=1580 MHz go through. Leaving the two unwanted images in for the baseband converter could hurt the downconversion process if they saturate the amplifier.

5.2.3 The Baseband Converter

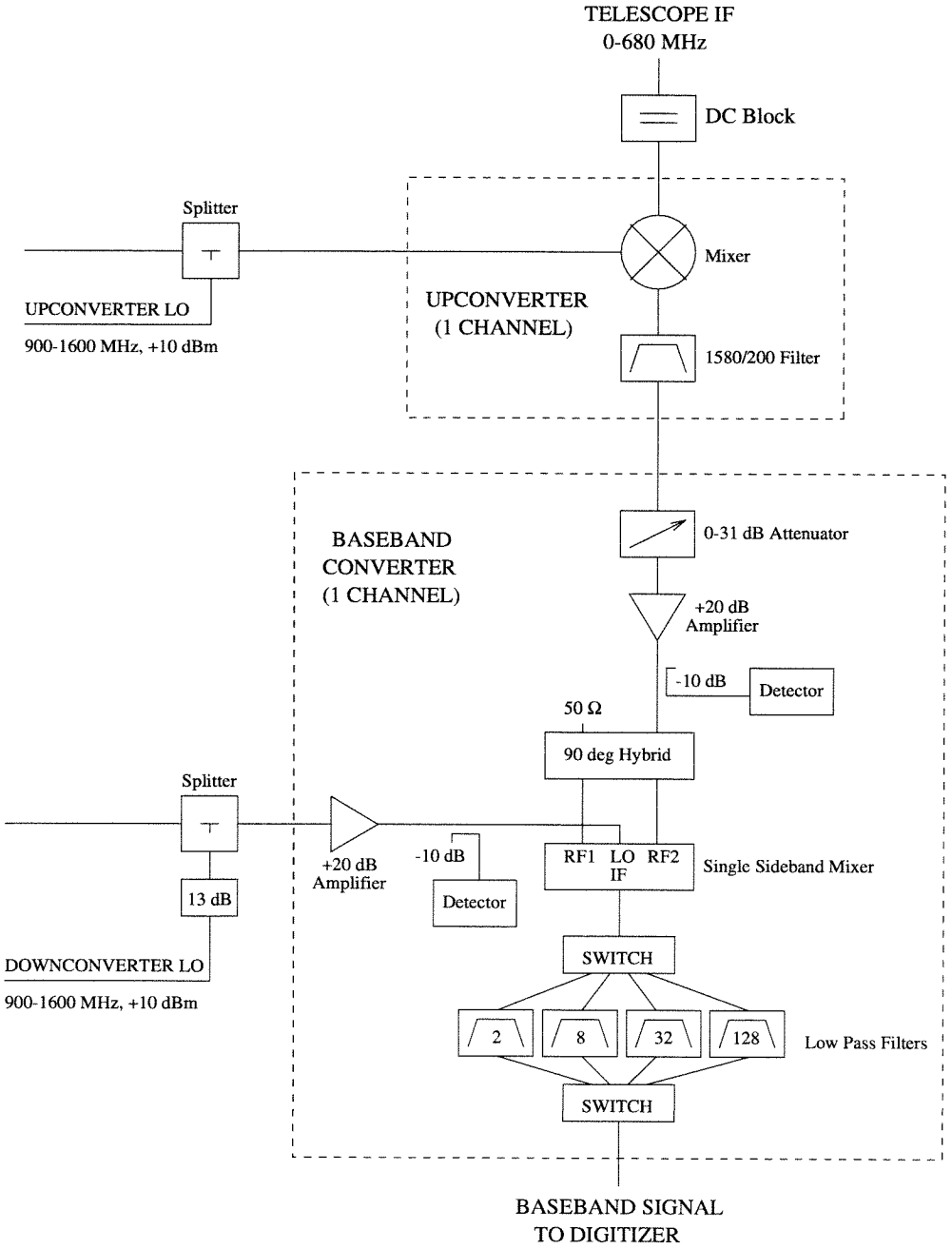
The two astronomical signals from the telescope, now with center frequencies of 1580 MHz, are passed from the upconverter to the Baseband Converter. The BBC comprises two sets of modules to amplify these input signals, mix them down to baseband and finally filter them down to the desired observing frequency. Figure 5.2 shows the layout of one BBC channel.

The amplifier consists of a computer controlled attenuator with a range of 0–31 dB followed by a fixed power amplifier, creating the effect of variable amplification. This amplifier is discussed at length below in §5.3, in relation to the required signal power levels.

The mixing in both channels is done with the help of an LO generator identical to the

Figure 5.2: The IF Subsystem

One of two channels in the IF subsystem. The telescope signal is mixed up to the 1580 MHz range for the baseband converter. Before going into the BBC the lower sideband is eliminated with a 200 MHz filter. In the BBC, the signal is attenuated and amplified before going into a single sideband mixer composed of a 90° hybrid and a quadrature mixer. The baseband signal is run through one of four filters to limit the total bandwidth, and is output into the digitizer. There are two power detectors in each IF channel, one for the LO and the other for the IF signal. The output voltages from the power detectors are displayed in the BBC front panel.



one used in the upconverter, and can generate all multiples of 1.25 MHz in the range 900–1600 MHz. The downconversion frequency $f_{\rm DOWN}$ defines the frequency that will translate to baseband. In general $f_{\rm DOWN} = f_{\rm USB} - {\rm BW}/2 = 1580 - {\rm BW}/2$, and the conversion is always done in USB mode. The left edge of the input signal goes from $1580 - {\rm BW}/2$ all the way down to DC, or 0 MHz.

The final stage in the BBC is to filter the baseband signals to one of four possible observing bandwidths: 2, 8, 32 and 128 MHz. The bandwidth is chosen by selecting one of four filters by means of analog switches. The filters are active and amplify the signals differently, so that the power at their outputs is identical and independent of bandwidth. As we will see below the inputs have constant power density and the outputs have constant total power, which requires that the smaller bandwidth filters have larger amplification.

The dual baseband converter (BBC) was originally cloned from the OVRO Millimeter Correlator. We have installed a computer control board to allow us to set the attenuation and output bandwidth from the observing program. The outputs from the power detectors are not accessible from the computer, but at least appear in the front panel of the BBC.

5.2.4 Sample Calculations of the Sky Frequency

Given the large number of upconversions and downconversions, it is sometimes difficult to know what the observing frequency really is. It is, of course, extremely important to know exactly what the observing frequency is and more so for pulsar timing, so here we present a few examples. Most of the difficulty, however, comes not from the number of conversions but from the limitations of the LO generators in the upconverter and the BBC. The quantization makes it impossible to know *a priori* what the sky frequency will be, and instead what one must do is choose an approximate sky frequency first, then work backwards from baseband to obtain the actual, exact sky frequency.

In the simplest case, assume we are using the P-band receiver to observe at 436 MHz, with a bandwidth of 32 MHz. According to table 5.1, the IFs for this receiver come down to the telescope room at the observing frequency. Clearly $f_{\rm UP}$=1144 MHz would upconvert 436 MHz to 1580 MHz, and $f_{\rm DOWN}$=1564 MHz would downconvert that to baseband, allowing us to preserve the goal $f_{\rm SKY}$=436 MHz.

Things are not that simple, unfortunately, because neither 1144 MHz nor 1564 MHz are multiples of 1.25 MHz: the closest multiples are 1143.75 MHz and 1563.75 MHz. Thus the center frequency at baseband, namely ${\rm BW}/2 = 16$ MHz, translates to the sky frequency

$$\begin{aligned} f_{\rm SKY} &= {\rm BW}/2 + f_{\rm DOWN} - f_{\rm UP} \\ &= 16 + 1563.75 - 1143.75 = 436 \text{ MHz}. \end{aligned}$$

In this example it was a coincidence that the two errors of 0.25 MHz cancelled out.

An observing run at 2400 MHz with 128 MHz of bandwidth would be slightly more

Table 5.2: Sample Setups for the IF Chain

Some of the most common observing configurations used at Parkes. Each line corresponds to a possible frequency setup, starting with the sky frequency at the receiver $f_{\rm SKY}$ and ending at baseband. The downconversion from $f_{\rm SKY}$ to $f_{\rm IF}$ occurs at the receiver, while the IF signal is processed down to baseband inside the FPTM. Center frequencies are accompanied by bandwidths, while LO frequencies stand alone, with the $\otimes$ symbol indicating that they are mixed in with the astronomical signal. Bandwidths are marked with a * if they are LSB with respect to the sky frequency. All frequencies are in MHz.

$f_{\rm SKY}$	$f_{\rm LO}$ $\otimes$	$f_{\rm IF}$	$f_{\rm UP}$ $\otimes$	$f_{\rm USB}$	$f_{\rm DOWN}$ $\otimes$	$f_{\rm BB}$
436.50/8	—	436.50/8	1143.75	1580.25/8	1576.25	4.00/8
436.00/32	—	436.00/32	1143.75	1579.75/32	1563.75	16.00/32
660.50/32	—	660.50/32	918.75	1579.75/32	1563.75	16.00/32
1520.29/32	1940.04	419.75/32*	1160.00	1579.75/32*	1563.75	16.00/32*
1519.79/128	1940.04	420.20/128*	1160.00	1580.25/128*	1516.25	64.00/128*
2400.25/128	2090.00	310.25/128	1270.00	1580.25/128	1516.25	64.00/128
4779.75/128	5100.00	320.25/128*	1260.00	1580.25/128*	1516.25	64.00/128*

complicated. We will step through the calculation slowly for the sake of correctness, if not clarity. From table 5.1 we see that the telescope LO is set to $f_{\rm LO}$=2090 MHz. The entire IF chain performs the following operation on the sky center frequency:

$$\begin{aligned} &f_{\rm SKY} - f_{\rm LO} + f_{\rm UP} - f_{\rm DOWN} - {\rm BW}/2 = 0, \quad \text{or} \\ &f_{\rm SKY} = {\rm BW}/2 + f_{\rm DOWN} - f_{\rm UP} + f_{\rm LO}. \end{aligned} \tag{5.1}$$

We also have the requirements from the two LO generators,

$$\begin{aligned} f_{\rm SKY} - f_{\rm LO} + f_{\rm UP} &\approx 1580 \text{ MHz}, \quad \text{and} \\ f_{\rm DOWN} + {\rm BW}/2 &\approx 1580 \text{ MHz}. \end{aligned} \tag{5.2}$$

Substituting the known or desired frequencies into equations (5.2),

$$\begin{aligned} 2400 - 2090 + f_{\rm UP} &\approx 1580 \text{ MHz}, \quad \text{and} \\ f_{\rm DOWN} + 64 &\approx 1580 \text{ MHz}. \end{aligned}$$

which are best satisfied with $f_{\rm UP}$=1270 MHz and $f_{\rm DOWN}$=1516.25 MHz, both multiples of 1.25 MHz. Now substituting into equation (5.1) we have

$$\begin{aligned} f_{\rm SKY} &= {\rm BW}/2 + f_{\rm DOWN} - f_{\rm UP} + f_{\rm LO} \\ &= 64 + 1516.25 - 1270 + 2090 = 2400.25 \text{ MHz}. \end{aligned}$$

Thus the sky frequency will be 2400.25 MHz instead of 2400 MHz.

There is yet another level of complication introduced when the telescope downconverts

the sky frequency in LSB mode. In that case equation (5.1) becomes

$$\begin{aligned} (f_{\rm LO} - f_{\rm SKY}) + f_{\rm UP} - f_{\rm DOWN} - {\rm BW}/2 = 0, \quad \text{or} \\ f_{\rm SKY} = f_{\rm LO} + f_{\rm UP} - f_{\rm DOWN} - {\rm BW}/2, \end{aligned} \tag{5.3}$$

and equations (5.2) are equally modified:

$$\begin{aligned} f_{\rm LO} - f_{\rm SKY} + f_{\rm UP} &\approx 1580 \text{ MHz}, \quad \text{and} \\ f_{\rm DOWN} + {\rm BW}/2 &\approx 1580 \text{ MHz}. \end{aligned} \tag{5.4}$$

Lower sideband downconversion is used with the L-band receiver. Assume that the L-band receiver is up and that we want to observe at about 1520 MHz with 128 MHz of bandwidth. This receiver downconverts the observing sky frequency with $f_{\rm LO}$=1940.04 MHz. As a result, equations (5.4) become

$$\begin{aligned} 1940.04 - 1520 + f_{\rm UP} &\approx 1580 \text{ MHz}, \quad \text{and} \\ f_{\rm DOWN} + 64 &\approx 1580 \text{ MHz}, \end{aligned}$$

from which we obtain $f_{\rm UP}$=1160 MHz and $f_{\rm DOWN}$=1516.25 MHz. Plugging these values into equation (5.3) we get

$$\begin{aligned} f_{\rm SKY} &= f_{\rm LO} + f_{\rm UP} - f_{\rm DOWN} - {\rm BW}/2 \\ &= 1940.04 + 1160 - 1516.25 - 64 = 1519.79 \text{ MHz}. \end{aligned}$$

The examples above are summarized in table 5.2. The observer does not actually have to set the three LO frequencies by hand, or even compute them every time. The telescope LO frequency $f_{\rm LO}$ is set by observatory staff during the receiver change and $f_{\rm UP}$ and $f_{\rm DOWN}$ are calculated and set automatically by the FPTM software prior to every observation. However, the observer does need to enter the *desired* IF center frequency $f_{\rm IF}$ into the FPTM, from which it calculates $f_{\rm UP}$ and $f_{\rm DOWN}$ and returns the *actual* IF center frequency. With this final $f_{\rm IF}$ the observer can compute the sky center frequency $f_{\rm SKY}$.

Once the sky frequency $f_{\rm SKY}$ has been chosen and is known exactly, it can be easily changed by modifying $f_{\rm UP}$. This means that $f_{\rm SKY}$ can be changed by multiples of 1.25 MHz from the FPTM by just typing a command. Of course, one can also change $f_{\rm SKY}$ by changing the $f_{\rm LO}$ that the Hewlett-Packard generator produces in the control room, and this change is not subject to the quantization limitation.

Table 5.3: Signal Levels

The signal levels in the FPTM, given in terms of the power density from the IF up to the BBC and in terms of total power thereafter. The IF power density can be different from −2 dBm/GHz as long as the attenuation in the BBC is changed accordingly.

Total Power dBm	Power Density dBm /GHz	Power Density dBm /300 kHz	Test Point
	−2	−37	IF inputs before upconversion
	−10	−45	Upconverter output, after filtering
	−25	−60	In BBC, after 15 dB attenuation
	−5	−40	In BBC, after 20 dB amplification
−10			BBC output, before 10 dB filter
−20			ALC input, after 10 dB filter
+19			ALC output, before 6 dB filter
+13			Digitizer input, after 6 dB filter

5.3 Power Levels

Signal Power Levels

The input signal levels are conditioned to the proper spectral power density, which in the current configuration is around −2 dBm/GHz. It is given in terms of spectral power density and not total power because the BBC has active output filters that ensure the same output power levels regardless of bandwidth. This means that IFs of 2, 8, 32 and 128 MHz of bandwidth should have total powers of −29, −23, −17 and −11 dBm respectively.

The power levels in the upconverter and downconverter are not very critical; it is the digitizer that requires precise and stable power levels regardless of the signal bandwidth. The levels must be stable because the digitizer threshold levels are constant (see chapter 6). In this section we follow the power levels in the IF rack, from the very input through upconversion, filtering at 1580 MHz, variable attenuation, constant amplification and finally baseband filtering in the BBC. A summary is provided in table 5.3.

At Parkes, spectral power densities are measured "per 300 kHz" instead of "per GHz," and in those units the ideal IF power density is −37 dBm/300 kHz. It is not necessary that it be *exactly* −37 dBm/300 kHz because the BBC has a 0–31 dB computer controlled attenuator to adjust the level. However, −37 dBm/300 kHz is the input power level that corresponds to an attenuation of 15 dB at the BBC, and with which the FPTM can in principle handle IF signals that are as many 15 dB above or below −37 dBm/300 kHz. Having the 15 dB of leeway gives a lot of flexibility in the observations. The attenuation can be set from the computer or by hand by means of switches on the BBC front panel.

Assuming that the input level is −37 dBm/300 kHz, upconversion and filtering at

1580 MHz add some 8 dB of attenuation due to losses and the BBC input attenuator adds another 15 dB, bringing the power density down to −25 dB/GHz at the input of the BBC downconverter. This is very close to the nominal value of −26 dBm/GHz. The reason why we don't work backwards and quote the ideal input power density as −38 dBm/300 kHz is because the exact attenuations are not 8 dB and 15 dB, and 1 dB of error does not affect the correlation function significantly.

After providing 15 dB (nominally 0–31 dB) of attenuation at 1580 MHz, the BBC amplifies the signal by 20 dB, downconverts it to baseband and filters it to the observing bandwidth. The four baseband filters (2, 8, 32 and 128 MHz bandwidth) are active and amplify by different amounts so that identical power densities at the input turn into identical total powers at the output. The baseband signals have a total power of −10 dBm regardless of their bandwidth. Thus, we have shifted from constant power density (as required by the downconverter) to constant total power (as required by the digitizer).

There is a power detector for each channel in the BBC. The detector is coupled to the signal path with a 10 dB coupler right after the 20 dB amplifier. Its output is nonlinear with respect to the signal power, but is easily calibrated with a power meter. The output voltage from both power detectors is displayed in the BBC front panel for convenience. Unfortunately, it cannot be read from the computer, but it is still useful during the setup and for debugging.

As we will see later, from the BBC the two baseband signals go to the digitizer module, where they are amplified again by the ALC units. For historical reasons they require an input power of −20 dBm, which forces us to install 10 dB pads between the two BBC outputs and the ALC inputs. These pads actually play an important role when they are at the ALC inputs rather than the BBC outputs, as there is significant noise pickup in that cable and they attenuate the pickup noise to a negligible level. The ALC amplifies the baseband signals from −20 dBm to +19 dBm, which then go through 6 dB pads and into the digitizers proper. The ideal total power at the digitizer is +13 dBm (or 1 V_{RMS}) per input. The outputs of the digitizers are ECL levels, for which power levels do no longer apply. The same is true for the correlator and the pulsar backend.

Local Oscillator Power Levels

Both the upconverter and the downconverter need LO powers of +10 dBm at the input. In the upconverter the LO is split before being fed into the mixers as +7 dBm LOs. In the BBC a 13 dB attenuator pad reduces the LO power to −3 dBm, which is further reduced to −6 dBm after splitting the signal for the two single sideband mixers.

Chapter 6

The Digitizer Module

6.1 Introduction

The two baseband signals from the IF rack are digitized and distributed before being sent to the correlator boards. The digitizer must sample each input signal at a rate of 256 MHz, twice the maximum bandwidth. In the current setup it always samples at this rate regardless of the baseband signal bandwidth. The essential advantage of fixed frequency sampling is that the digitizer always works in the same regime, with the same clock rates, delays, noise and self-made interference, and only needs to be debugged at that one high frequency of operation. For bandwidths smaller than 128 MHz, this means that the input signal will be oversampled and that adjacent samples will not be independent, but this is not a problem since only every other (or every nth) sample is correlated.

The digitizer was designed and built by Steve Padin (Padin and Ewing 1989) for the Owens Valley Millimeter Array, and is now in operation at the Caltech Owens Valley Radio Observatory (Padin *et al.* 1991). It can digitize two channels or baseband signals with a resolution of 2 bits at a rate of 256 Msamples/s. The module contains two automatic level controllers, a dual digitizer circuit and two threshold monitors. We will refer to the actual digitizer circuit as "the digitizer," unless otherwise specified. The digitizer takes two +13 dBm signals, usually from the two states of polarization, and produces two digital data streams. The ALCs merely amplify the baseband signals from the BBC output to the level of +13 dBm for the digitizer and act as power limiters for protection, should the input signal be much above the nominal −20 dBm. The threshold monitors provide convenient outputs to monitor the performance of the digitizer.

6.2 The Digitizer ALCs

Upon entering the digitizer module, each baseband signal goes through an Automatic Level Controller (ALC). Originally, in the Millimeter Interferometer, the ALCs were variable am-

plifiers in a feedback loop that would raise the input signal to a constant level of +13 dBm, the power level required by the digitizer thresholds. Keeping the digitizer input at a constant power level is a good idea, mainly to protect the digitizer from accidental burnout, but also to maintain the analog signals at the ideal level with respect to the *fixed* digitizer thresholds. This results in balanced digital data streams that permit the correlator to operate at the maximum efficiency.

In contrast with interferometry, pulsar work requires that we keep track of the total power of the input signal as a function of time, and therefore automatic attenuation with time scales shorter than or comparable to the pulsar period would be detrimental. As a result, we modified the ALCs and turned them into constant factor amplifiers. This modification can be dangerous because, should the input power surge, the ALCs would amplify the increase in power and might damage the digitizer. Thus we made sure that the ALCs saturated at a level that is high for the digitizer but would not destroy it, just like a power limiter. A better solution would be to restore the ALCs to their original variable amplification, but increase the feedback time constant. A time constant of 5–10 seconds would let any pulsar through, yet limit an accidental power surge to less than a minute and protect the digitizer chip set to some extent.

Each ALC contains a diode as a power detector in the signal path. Its (nonlinear) output voltage is displayed in the digitizer front panel, and is calibrated for debugging and for monitoring the digitizer conditions prior to each observation.

6.3 The Digitizer Proper

For each one of two analog +13 dBm baseband signals the digitizer performs 4-level (2-bit) quantization at a rate of 256 MHz. The 2 bits are called **sign** (S) and **magnitude** (M) and are generated by comparing the input signal with three thresholds. The sign bit is produced with a threshold of 0 Volt whereas the magnitude bit is produced with two thresholds symmetric across 0 Volt. If these thresholds are $-v_0$ and $+v_0$, then the magnitude bit will be set HIGH whenever the input signal voltage is less than $-v_0$ or more than $+v_0$ (see table 6.1). All the digital outputs are ECL and need to be terminated accordingly when being tested. Proper termination methods are described in chapter E.

The four levels coded by the two digital bits are SM={11,10.00.01}, where the first bit is the sign and the second is the magnitude. They are interpreted by the correlator as having the integral values $\{-3, -1, +1, +3\}$ respectively, and are correlated according to these values.

The sign and magnitude bits for the two channels are produced at a rate of 256 MHz. The rate is determined by the **digitizer clock**, CCLK. The digitizer module will function correctly with CCLK up to 400 MHz, but the correlator boards which follow cannot keep

Table 6.1: Digitizer Thresholds and Output Values
The digitizer performs 4-level quantization on the input signals. The resulting 2 bits are called **sign** (S) and **magnitude** (M) and are determined by comparison of the input signal with the 3 thresholds $-v_0$, 0 and $+v_0$. The correlator later interprets the four possible levels as having integer values of $-3, -1, +1$ and $+3$, which maximizes the SNR.

Voltage	S M	Value
$v(t) < -v_0$	1 1	-3
$-v_0 < v(t) < 0$	1 0	-1
$0 < v(t) < +v_0$	0 0	$+1$
$+v_0 < v(t)$	0 1	$+3$

up with the data at rates above 260–270 MHz.

6.4 The Threshold Monitors

For the nominal input power of +13 dBm (1 V_{RMS}), the three thresholds in each channel are normally set to −0.9 V, 0 V and +0.9 V. The actual value of each threshold can be changed with a small screwdriver by tweaking a ten-turn potentiometer through a side wall of the digitizer module. The six thresholds remain very stable and should be adjusted as seldom as possible.

At every cycle of CCLK the digitizer compares the input signal with the three thresholds: $-v_0$, 0 and $+v_0$. Although the three decision bits are combined into two output data bits, they are also provided as outputs for monitoring. Due to the high data rates they are first divided by 16 and buffered, so that the maximum output rate in any of the **threshold monitors** is just 16 Mct/s.

The threshold monitor outputs make it possible to see how the digitizer is performing, with the use of a simple frequency counter. Such a counter is built into the digitizer module, and any of the six threshold monitor rates can be displayed in the front panel. The expected count rates during normal operation are described below.

In general it is not necessary to check the threshold monitor outputs unless there is already indication that one of the digitizers is having trouble. At the beginning of every run we do some test observations of the artificial pulsar generator, a test which fails clearly when the digitizers are not working properly.

6.5 Count Rates in the Digitizer Module

When the appropriate IF power level is given to the digitizer (+13 dBm), the number of hits above and below each threshold is statistically determined. Likewise, the statistics of the sign and magnitude bits out of the digitizer follow well-determined statistics, as do the

results of the correlation before and after integration and the numbers read out from the ramadder boards after further integration and folding at the pulsar period. In this section we summarize what the count rates at the digitizer should be for help in the debugging process.

The threshold monitors compare each analog baseband signals against three stable analog thresholds. In the ideal situation the input signal has a power level of +13 dBm, which corresponds to 1 V_{RMS}, and the three thresholds are −0.9 V, 0 V and +0.9 V. In this setup the sign bit will be HIGH exactly half the times, which with a clock of 256 MHz corresponds to 128 Mct/s. The magnitude bit is only high 37% of the time, and its count rate will be 94.2 Mct/s.

It would seem *a priori* that the ideal thresholds would produce a magnitude bit HIGH 50% of the time like the sign bit, since that would make all four levels equally likely and maximize the information kept about the input signal. However, what we want to maximize is not our knowledge of the input signal but of the correlation function. When the signal-to-noise ratio of a real signal with a $V_{RMS} = \sigma$=1 Volt is optimized in the correlation function as a function of the symmetric thresholds $-v_0$ and $+v_0$, we obtain the value v_0=0.9 V for the optimal threshold, versus v_0=0.67 V for the 50% HIGH magnitude bit. This of course depends on the values the correlator assigns to the digitizer samples, and on the multiplication table that the correlator uses. Very detailed analyses of this issue are given in Cole (1968) and Cooper (1970), and a clear and comprehensive summary can be found in Thompson, Moran and Swenson (1991).

If we accept that the optimal thresholds are given by v_0=0.9 Volt, the probabilities that the signal is above or below each threshold are easily calculated.

$$P(-v_0 < v(t) < v_0) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-v_0}^{+v_0} e^{-\frac{x^2}{2\sigma^2}} dx = \mathrm{erf}\left(\frac{v_0}{\sigma\sqrt{2}}\right) = \mathrm{erf}(0.636) = 0.632.$$

from which the other probabilities follow by symmetry. Hence $P(v > v_0) = 0.184$ and the magnitude bit will be HIGH 37% of the time, as mentioned above. The expected rates are summarized in table 6.2.

It is difficult to find a standalone counter for ECL rates above 100 MHz, and for this reason the digitizer includes a set of dividers that produce a slower output for the sole purpose of debugging. These circuits, one per channel, are called the threshold monitors, and produce ECL output rates 1/16th of those from the digitizer. Thus the zero threshold monitor will have an output rate of 8 MHz and the $-v_0$ and $+v_0$ thresholds rates will be 2.94 MHz or 13.06 MHz, depending on whether we are looking at the hits *above* or *below* the threshold.

Table 6.2: Expected Threshold Monitor and Digitizer Rates
The rates that are expected from the threshold monitors and the digitizer when v_0=0.9 Volt and σ=1 Volt. The sign bit is obtained directly from the 0 V threshold, while the magnitude bit is a combination of the $-v_0$ and v_0 threshold comparisons. The rates are given in MHz.

Event	Prob.	Rate
$v(t) < -v_0$	0.184	2.94
$-v_0 < v(t)$	0.632	13.06
$v(t) < 0$	0.500	8.00
$0 < v(t)$	0.500	8.00
$v(t) < +v_0$	0.632	13.06
$+v_0 < v(t)$	0.184	2.94
S=0	0.500	128.00
S=1	0.500	128.00
M=0	0.632	161.76
M=1	0.368	94.24

6.6 Digitizer Tests

There are two tests of the digitizer for which the threshold monitors are very helpful. First, when a baseband signal with Gaussian statistics and the appropriate power level is input into the digitizer, the threshold monitor rates should be 8 MHz and ~3 MHz. If they aren't then the thresholds must be adjusted. This adjustment can be done with a small flat head screwdriver, by tuning the ten-turn potentiometers that control the threshold voltages. There is one ten-turn pot for each threshold level, and they can be accessed through holes in the side of the digitizer module. Note: be absolutely convinced that the levels *must* be adjusted before proceeding, and extremely careful when adjusting them. The exact threshold levels are not very critical because the correlator signal-to-noise ratio does not depend on them too strongly, so there is little point in getting the exact theoretical count rates. Threshold rates in the ranges 7.9–8.1 MHz and 2.7–3.1 MHz are perfectly acceptable.

The second test is to feed the digitizer a 32 MHz sine wave of also +13 dBm. If this sine wave is phase locked to the digitizer clock CCLK, then we know exactly how the output bits must run (they correspond to the circular series $-3, -1, -1, +1, +3, +3, +1$). Whereas the first test was ideal for a Gaussian random signal and for testing the thresholds, the second is a better test of the digitized data outputs. Sometimes an output buffer is burnt out in the digitizer module and the correlator does not get a particular bit, but the threshold monitor outputs look fine. In that case this test is essential. The solution is either to replace the output buffer (a surface mount ECL chip), or to just use from another output (there are eight identical outputs for each bit) and label the bad output for future reference.

Chapter 7

The Correlator

7.1 Introduction

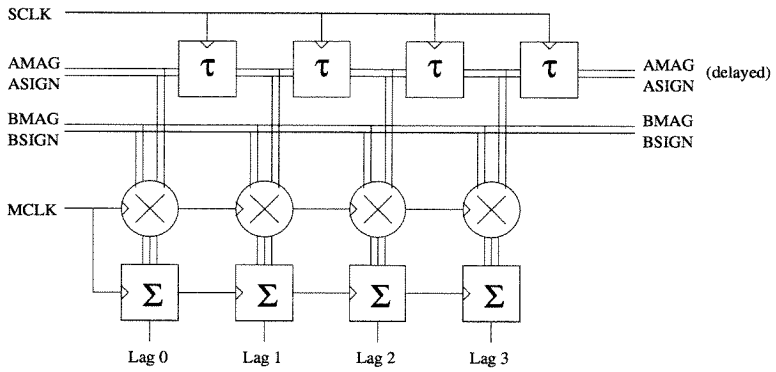
The correlator is the computational heart of the pulsar timing machine. It does most of the on-line computing by calculating as many as 512 lags of the auto- and crosscorrelation functions in real time. It was designed and built by Martin Ewing at Caltech for the Owens Valley Millimiter Array (Ewing 1989b) and it is the basis of the current array (Padin *et al.* 1993). In the FPTM we use one baseline's worth of correlator boards, *i.e.*, a single correlator cage, with a few changes to improve the computer interface and data readout. The correlator cage contains eight boards working in parallel and one computer to control them.

The data from the digitizer is passed from one board to the next in a daisychain, while the clocks are all identical copies of the same clock signals from the clock generator. Each board calculates 64 lags of a correlation function, either one of the autocorrelations or crosscorrelations depending on the observing configuration. When all the boards are chained they can evaluate up to 512 lags of the same correlation function, but in general the 512 lags are broken into two autocorrelations of 256 lags, or two autocorrelations of 128 lags and two crosscorrelations of 128 lags. The particular configuration to be used in each observation is chosen through the software and does not need any physical swapping of wires.

The correlator boards are based on two custom-made chips: the **High Speed Correlator Chip**, a 2500-gate ECL macrocell masked gate array designed at Caltech, and a 32-channel, 18-bit synchronous accumulator chip designed at JPL. The latter is normally called the **JPL Accumulator**.

Figure 7.1: The Heart of the Correlator

Each lag in a correlator is evaluated from the two digitizer data streams by a process of delay, multiplication and accumulation. Of the two data streams, one is passed from one lag to the next without delay while the other suffers a delay of one clock cycle. The delay τ determines the total bandwidth of the correlator, the number of lags the frequency resolution, and the accumulator readout interval the time resolution of the correlator.



7.2 The Correlator Clocks

The essential correlator is shown in figure 7.1. Each lag consists of a delay unit, a multiplier and an accumulator. In a digital correlator each delay unit is implemented with a holding register, and the lag delay τ is set by the **shift clock** SCLK, which clocks samples into and out of the holding registers. The other fundamental clock is the **multiply clock** MCLK and it determines when the two samples in each lag are multiplied and accumulated. The shift and multiply clocks must have precise frequency and phase relationships so that the two samples in each lag remain stable while they are being multiplied. In most correlators the two clocks have the same frequency, set by the lag delay, and their relative phase is that required by the particular hardware implementation.

The total bandwidth of a correlator is given by $1/2\tau$ and its frequency resolution by $1/2N\tau$, where N is the number of lags. To avoid aliasing of the input radio signal, the correlator bandwidth must be equal to or larger than the bandwidth of the input signal from the digitizer. Given the four possible bandwidths in the IF subsystem, 2, 8, 32 and 128 MHz, the shift clock must run at twice the observing bandwidth, at one of 4, 16, 64 and 256 MHz (see table 7.1).

For observing bandwidths of 2, 8, 32 and 128 MHz, the digitizer will oversample the analog data stream by factors of n=64, 16, 4 and 1, but it will still produce two digital data streams at a rate of 256 MHz. Adjacent samples from the digitizer will not be independent, and for this reason the correlator only needs to use every nth one. Hence the shift clock is matched to the observing bandwidth, not the bandwidth of the digitizer. Nevertheless the multiply clock always runs at 256 MHz, effectively correlating each independent pair of samples n times before the shift clock brings in a new pair. The time between two

Table 7.1: Correlator Clocks and Bandwidth

The three clocks CCLK, SCLK and MCLK (digitizer, shift and multiply) determine the timing and bandwidth of the correlator. n is the number of times that each independent digitizer sample is correlated in a given lag unit. The bandwidth in the last column is the *correlator* bandwidth, which should be equal to or larger than the observing bandwidth (see text).

CCLK MHz	SCLK MHz	MCLK MHz	n	τ ns	BW MHz
256	256	256	1	3.9	128
256	64	256	4	15.6	32
256	16	256	16	62.5	8
256	4	256	64	250.0	2

accumulator dumps determines the time resolution of the correlator.

As long as the digitizer, shift and multiply clocks are locked in frequency and have the proper phase relationships, this approach presents several advantages. First, since SCLK is variable, the correlator bandwidth can be matched to the observing bandwidth and the frequency resolution of the correlator is used to its fullest. Second, the time between accumulator dumps can be changed to match the desired time resolution for the correlator. Next, since MCLK is constant, the correlator output rates are independent of bandwidth and therefore the rest of the hardware downstream from the correlator can operate at a single frequency. And last but not least, having CCLK and MCLK fixed at 256 MHz eliminates most of the correlator the problems that come with variable frequencies, delays and self-made noise.

7.3 The Correlator Chip

The High Speed Correlator Chip (Ewing 1989a) was designed by Martin Ewing at Caltech and built by Motorola in 1986. It takes two high speed clocks as well as two 2-bit digital input streams, and outputs the first 8 lags of the correlation function and the two original data streams, one of them delayed by 7 or 8 clock cycles for cascading to another chip.

All input and output signals are ECL, and when the clocks run at 256 MHz each chip dissipates some 8 Watt of power. For this reason a heat sink is required on each chip and air flow is forced through the heat sink. Since the chips can be damaged if there is no ventilation, a temperature sensor turns the power supply off if the chips get too hot.

The two clocks going into the correlator chips are the shift clock SCLK and the multiply clock MCLK from the clock generator. The shift clock determines the time delay between adjacent lags and defines the total bandwidth. From the sampling theorem SCLK must run at least at twice the bandwidth of the original IF signal, e.g., SCLK=256 MHz when the IF bandwidth is 128 MHz or less.

Table 7.2: Multiplication Table

Table *A* shows the basic multiplication table of the correlator when the two bits from each digitizer channel are interpreted as representing values of $-3, -1, +1$ and $+3$. Table *B* shows the modified multiplication where the low level (or inner) products are dropped. Table *C* shows the final multiplication table with a bias to make the result of the correlation always positive.

A

×	−3	−1	+1	+3
−3	+9	+3	−3	−9
−1	+3	+1	−1	−3
+1	−3	−1	+1	+3
+3	−9	−3	+3	+9

B

×	−3	−1	+1	+3
−3	+3	+1	−1	−3
−1	+1	0	0	−1
+1	−1	0	0	+1
+3	−3	−1	+1	+3

C

×	−3	−1	+1	+3
−3	6	4	2	0
−1	4	3	3	2
+1	2	3	3	4
+3	0	2	4	6

The multiply clock must be at least as fast as the shift clock, and in the FTPM it is kept constant at 256 MHz. The multiplied data is added into an accumulator and read out at a slower rate. It is the accumulation time that determines the finest time resolution of the correlator. If all the lags are read once each accumulation time (*i.e.*, an entire autocorrelation function is read), then a power spectrum can be calculated for that time interval. This power spectrum looks like a single time sample of a filterbank where each channel has a sync response (Blackman and Tuckey 1958). In the case of the pulsar timing machine, the finest time resolution we are interested in is a pulsar period phase bin, a much larger time that can be integrated in the accumulator chips, so further accumulation of the correlation data is performed in the pulsar backend.

The HSCC calculates the correlation function for each lag according to the multiplication table in table 7.2. One such product is calculated from the two input stream in every lag, in every cycle of MCLK. The resulting products, of values ranging from 0 to 6, are added into the 6-bit accumulator and the overflow bit is read at a rate of 32 MHz. Therefore, the accumulator output is serial and the lag value is proportional to the count rate. Reading the overflow bit of the accumulator amounts to dividing the accumulator count rate by 2^6, so the maximum count rate is $6/2^6 * 256$=24 Mct/s on a 32 MHz carrier signal (refer to §7.5).

The effective in-chip accumulation time is 41 ns, since that is the minimum time it would take to fill up the 6-bit accumulator. Further accumulation up to μs or ms (the size of a pulsar phase bin) is performed by the JPL accumulator chips or in the pulsar backend. The accumulator chips are not used during observations with the FPTM, although they are useful for debugging the correlator boards.

7.4 The JPL Accumulator Chip

In the OVRO Millimiter Array additional accumulation up to several ms is performed on the correlator boards with the JPL custom-made accumulator chips (LSI 1999). These accumulator chips contain 32 18-bit asynchronous counters apiece, and are read sequentially through a parallel bus that is interfaced into the VMEbus.

Since the 32 counters in each accumulator chip are read sequentially, and no two accumulator chips in the entire correlator crate can be read simultaneously, reading the lag data through the JPL accumulator chips takes as many VMEbus cycles as lags there are to be read. Even for an optimistic 100 ns per transfer, this corresponds to 50 μs for 512 lags, which is much larger than the smallest desired phase bin length for fast pulsars, a few μs. As a result, the JPL accumulators cannot be used for pulsar timing. There is yet another difficulty, the fact that the accumulators are disabled while they are being read. This implies a dead time of at least 50 μs in each phase bin, which makes matters even worse.

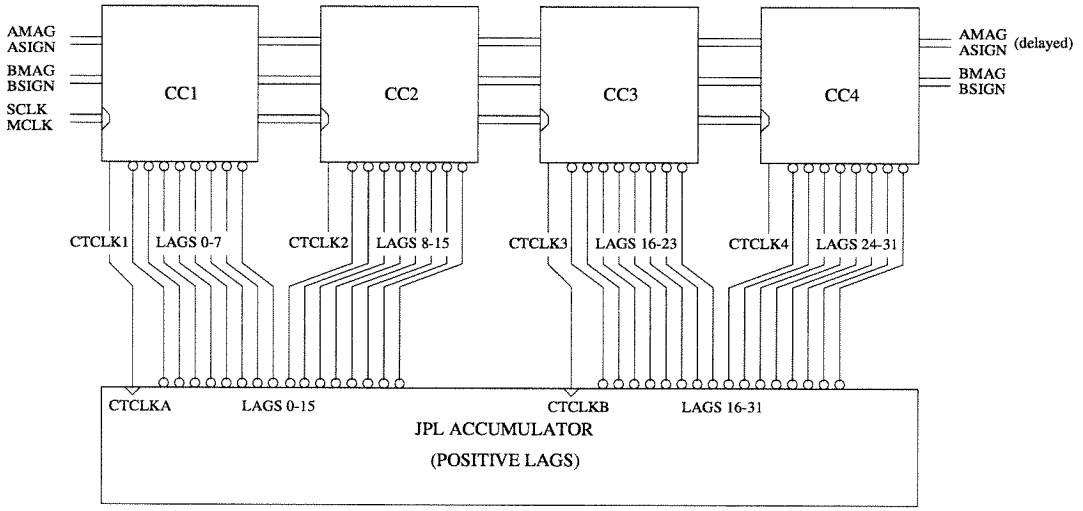
The 32 counters are arranged as two sets of 16, with a **count clock** input for each set. Thus the data are coded as logic levels to be sampled at the rising edges of the count clocks, and should be considered more like count enable inputs than count clocks themselves. The correlator chips were designed accordingly and provide count clock outputs in addition to the lag data outputs, but since each chip only correlates 8 lags, there is one count clock for every 8 lags. This means that the outputs from half of the correlator chips will be clocked into the accumulator counters by the count clocks from neighboring correlator chips, which can lead to timing problems. See figure 7.2.

It is almost hard to believe that, having solved all the timing problems with the 256 MHz clocks and data, we should now have trouble with the 32 MHz signals. These 32 MHz signals, however, are derived from the 256 MHz clocks in an independent manner, and could be out-of-phase with respect to one another by any number of 4 ns clock cycles. It is best to look at them as 256 MHz signals that only carry 32 Mbit per second, but which can still have transitions in any of the clock pulses of the 256 MHz clocks. The two chips whose lags share a count clock must have count clocks which are aligned, since one of them is discarded. The only way to align the 32 MHz count clocks is to issue a **multiply reset** pulse to the correlator chips after power-up. We generate such a pulse from the computer, through the digital interface.

In the FPTM the JPL accumulator chips are used exclusively for testing. If the correlator chips can be supplied with noise or artificial patterns, then the accumulator chips are useful because they allow the observer to test the correlator independently of the pulsar backend. Once the correlator crate is deemed to be working, a full check can be run on the pulsar backend itself. For the latter step a time-variable (and periodic!) source of noise is desirable, and we use a box we call the **artificial pulsar generator**.

Figure 7.2: The Correlator Outputs

The first four correlator chips of a board compute 32 positive lags and output the delayed digitizer data for the next board to continue correlating. The 32 positive lags can be accumulated on-board by one of the JPL accumulator chips, which uses the count clocks from every other correlator chip to drive its 32 asynchronous counters. Lags 8–15 and 24–31 are clocked by the "wrong" count clocks, those from lags 0–7 and 16–23. This means that the even chips must be perfectly synchronized with the odd chips, or else their lag data will be improperly counted. The negative lags (normally called lags 32–63) are handled in a like manner, by correlator chips 4–8 and the second JPL accumulator on the board. The 64 lags and the 4 count clocks from the correlator chips are made available to the pulsar backend by means of a front panel connector and ribbon cables.



The JPL accumulators are read into FIFOs in a burst to minimize the time lost to the observation, and the FIFOs are then read at leisure over the VMEbus, asynchronously.

7.5 Correlator Rates

The lag data from the correlator chips are sent to both the on-board accumulator chips and the ramadder boards in the pulsar backend, which also perform additional accumulation. In this section we calculate the data rates that should be expected during normal operation, which we will need later in §8.3.4 to calculate the optimal integration times for pulsar timing observations.

The lag data are calculated in the correlator chips through the process of multiplication and accumulation. From table 7.2 we see that each multiplication of two digitizer samples will result in a number between 0 and 6 to be added into the 6-bit accumulator, at a multiply clock rate of 256 MHz. Thus the minimum, zero correlation and maximum data rates are:

$$\{\min, 0, \max\} \quad — \quad 256 \text{ MHz } \frac{\{0, 3, 6\}}{2^6} = \{0, 12, 24\} \text{ Mct/s.}$$

Astronomical signals will normally show near-zero correlation with typical products

around the zero correlation value of 3, which is also the bias value of the multiplication table. Therefore, the count rates will be around 12 Mct/s. This is approximately true even for the zero lag of the autocorrelation function, where the correlation coefficient is ρ=1 and all the products are positive. From §6.5 on the digitizer rates we know the probabilities that a +13 dBm signal be above or below the three thresholds $-v_0$, 0 and $+v_0$, thus

$$\rho = +1 \quad \rightarrow \quad 256 \text{ MHz } \frac{0.63 \cdot 3 + 0.37 \cdot 6}{2^6} = 16.4 \text{ Mct/s}.$$

Until now we have neglected an important but little-known fact, that the lag data outputs from the correlator are active-LOW. Likewise, the JPL accumulator inputs expect the input data to be active-LOW. This fact is absent from the correlator documentation (Ewing 1989a) and caused us to design the ramadder boards to count ones when they should be counting zeroes. It is not a fundamental problem, but it results in effectively larger count rates and smaller integration times (because the memories fill up faster):

$$\{\min, 0, \max\} \quad \rightarrow \quad 256 \text{ MHz } \frac{8 - \{0, 3, 6\}}{2^6} = \{32, 20, 8\} \text{ Mct/s}, \quad \text{and}$$

$$\rho = +1 \quad \rightarrow \quad 256 \text{ MHz } \frac{8 - (0.63 \cdot 3 + 0.37 \cdot 6)}{2^6} = 15.6 \text{ Mct/s}$$

for the general case and for the zero lag of an autocorrelation respectively. This point will come up again in §8.3.4 when we look at the count rates in the ramadder boards.

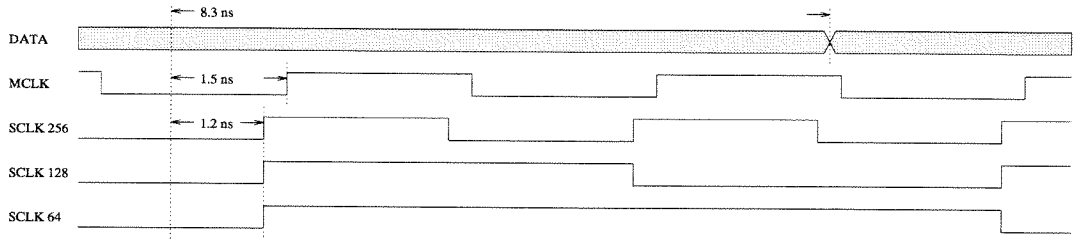
7.6 Clock and Data Issues

At 256 MHz an entire clock pulse is only one foot long, traveling at roughly a third of the speed of light, and its rising and falling edges can be each shorter than an inch. This means that the traces carrying clocks must be of the right length, or else a small timing error could generate spurious outputs and corrupt the data. Likewise, cables can only be swapped by new cables of the exact same length, or rather the same delay, as different cables of different materials sometimes have different propagation speeds and must be of unequal length in order to produce the same time delay in the signals they carry.

The timing relations that are most difficult to satisfy are between CCLK, MCLK and SCLK, mainly because the relationship between these clocks depends on the delays that the data suffers through the digitizer and the correlator. Some of these requirements can be satisfied by the right cable lengths between modules, as shown in figure 7.3.

Figure 7.3: Clock and Data Delays

This figure shows the relative timing expected between SCLK, MCLK and DATA at the correlator board inputs. All three are originally synchronous with the digitizer clock CCLK, but suffer delays before getting to the correlator boards. MCLK and SCLK are delayed by a fraction of a ns in the clock distribution module, while DATA is delayed by an additional 8.3 ns in the digitizer. The final timing relations given here are obtained by adding lengths of cable to SCLK and MCLK, to delay them by 1.2 and 1.5 ns with respect to DATA. In the end, the only relevant numbers are that MCLK lags SCLK by 0.3 ns and that DATA lags SCLK by 7.1 ns.



Chapter 8

The Pulsar Backend

8.1 Introduction

The main purpose of the pulsar backend is to fold the correlator data at the exact topocentric period of the pulsar into a number of phase bins, typically 1024 per period. The 512 lag data streams from the correlator are redirected into the appropriate phase bins in memory. For pulsar periods between a few milliseconds and a few seconds, this corresponds to accumulation times of a few μs to a few ms per phase bin, much larger than the 41 ns that the correlator chips can accumulate for, and even more so given that each integration typically consists of a large number of periods and that each phase bin gets visited once every period.

At the end of each integration, the 512 lag profiles, of 1024 phase bins each, are read into the computer and transformed into 512 pulse profiles, each at a particular sky frequency. The two-dimensional array of pulse profiles vs. frequency can be used to study the frequency structure of the pulsar signal and propagation effects in the interstellar medium, but is normally dedispersed into a single, high signal-to-noise ratio pulse profile, or rather one such pulse profile per polarization (see figure 8.1). The dedispersed pulse profiles are used to obtain times-of-arrival by fitting them to standard templates. Times-of-arrival can also be obtained for each individual frequency channel before dedispersion. This is preferred for strong pulsars, which can be seen in the individual frequency channels.

In this chapter we describe the pulsar backend, the hardware that folds the correlator lag data at the period of the pulsar. It is a triple-height VMEbus crate with a computer, a control board and a series of boards to do the actual folding. These boards are called the **ramadder** boards, short for "RAM-adder" since they add the incoming data to the previous data already in RAM memory. The crate also contains two PIO boards (appendix B), one for communication with the correlator crate and the other for communication with the LO generator, the downconverter and the digitizer through the digital interface.

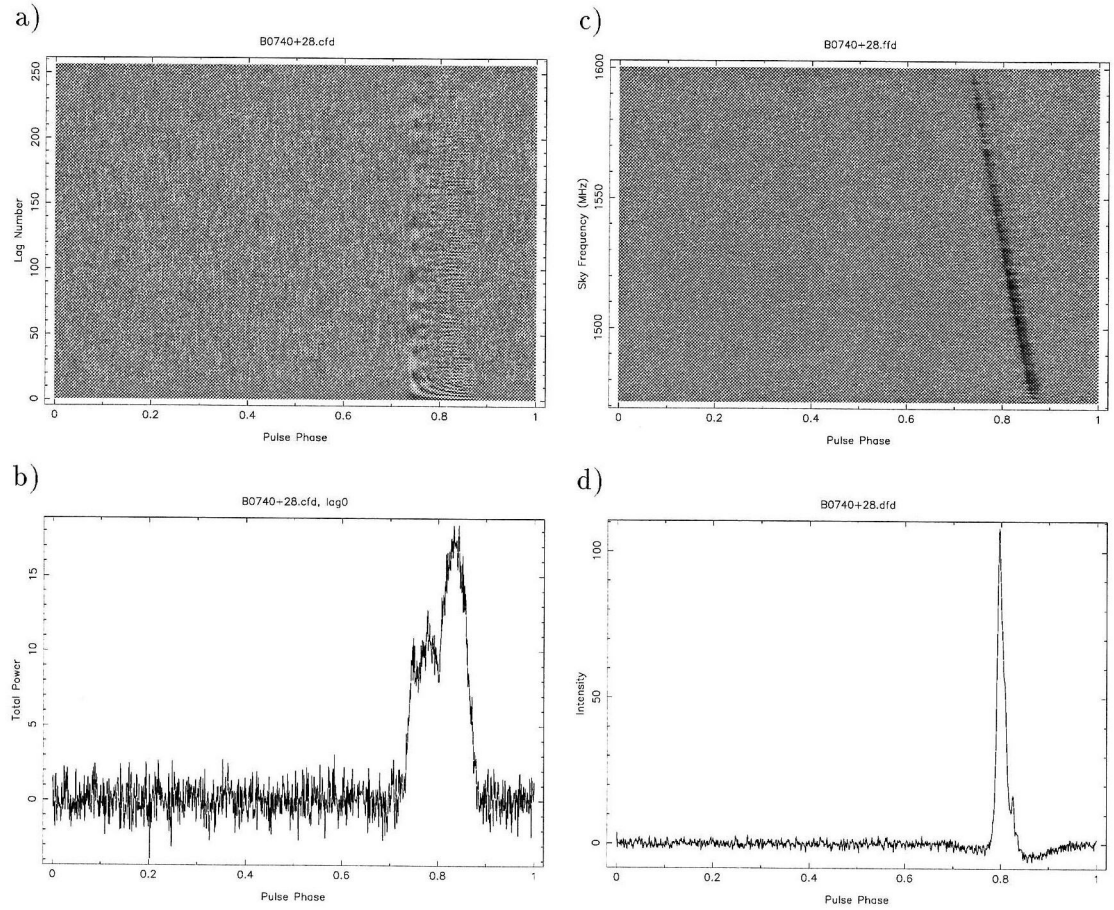
Unless otherwise specified, we assume that the observation uses all 512 lags and that

Figure 8.1: From Correlator Data to Pulse Profiles

The main steps in correlator data processing. The data shown here are from a 1 minute, 1536 MHz observation at Parkes of the pulsar B0740+28, with 256 lags per polarization folded into 1024 phase bins. Only one polarization is shown.

Panel *a* shows lag profiles as a function of lag number, the *correlator folded data.* It is essentially a raw dump of the correlator and the subject of this chapter. Lag 0 is shown alone in panel *b* and it is a direct measure of the total power of the input IF, the *dispersed* pulsar signal. Panel *c* shows pulse profiles as a function of sky frequency, the *filterbank folded data*, the data set that is easiest to visualize and interpret. It is obtained from panel *a* by doing a Fourier-style transform of the lag data in each phase bin. Finally, panel *d* shows the high signal-to-noise ratio pulse profile obtained by dedispersing panel *c* to the central frequency of 1536 MHz, the *dedispersed folded data.* One such dedispersed profile is obtained per state of polarization in each observation (perhaps with profiles of the other Stokes parameters), and they are the source of the times-of-arrival.

Note: plots *b* and *d* have arbitrary intensity units, and the data sets in panels *a* and *c* have been normalized so as to emphasize the pulsar features rather than the bandpass filter shape and other instrumental effects.



each period is divided into 1024 phase bins. Lags are numbered from 0 to 511 and phase bins from 0 to 1023.

8.2 Folding at the Pulsar Period

The ramadder boards process the data streams from the correlator in real time, folding each lag at the pulsar period into a large number of phase bins, typically 1024 (see figure 1.1). The computing power required to fold all 512 lags simultaneously is enormous, the total data rate from the correlator being 512 times 32 Mbit/s, or 2 Gbyte per second!

The data from the correlator boards arrives continuously in a 32 Mbit/s data stream per lag. In the ramadder boards each lag is redirected into a particular phase bin in memory depending on the pulsar period and the time since the beginning of the integration. After one period, when all the phase bins have been addressed, the process starts over again with the first phase bin. This time, however, the correlator data is added onto the data already in that phase bin from the previous period. Essentially, the accumulation of the lag data continues over many periods for the entire length of the integration (normally a few minutes), but as a function of phase bin. An integration will always last an exact number of pulsar periods so that all the phase bins are addressed the same number of times.

At the time when we designed the ramadder boards, TTL chips had trouble operating at 32 MHz, which forced us to slow the input data rate down to 16 Mbit/s by means of dividers. This amounts to throwing data away, but a calculation of the expected SNR shows that we can afford to do so. It is equivalent to having a 7-bit accumulator in the correlator chips instead of the current 6 bits.

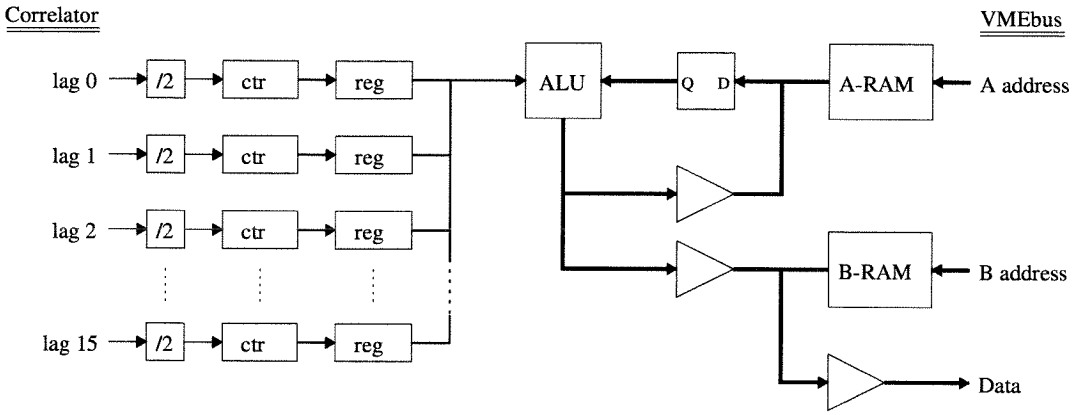
8.3 The Ramadder Boards

Each ramadder board processes 64 lags from the correlator, which greatly simplifies the wiring because it means that eight ramadder boards can be connected one-to-one with the eight correlator boards. Each lag is at this point no more than a 32 Mbit/s data stream where the count rate is proportional to the instantaneous correlation function at that lag. From the point of view of a ramadder board, it doesn't matter which lag from the correlator is which, and for all practical purposes they could be scrambled, as long as the computer can keep track of which one is which later on for the data reduction.

Each ramadder board performs a large number of operations per second while folding the lag data, but does so in a dumb way. It folds 64 lags in parallel, each lag entailing 20-bit data pathways and 14-bit addresses. Thus the boards are full of traces and packed with chips, but the logic operations they perform are simple. All the timing and control signals come from the **control board** in the same VMEbus cage, which is where the complicated

Figure 8.2: A Ramadder Board Section

Each ramadder board is divided in four sections, each of which processes a set of 16 correlator lags. The sections themselves process the 16 lags serially. Each section has 16 counters that are always accumulating correlator lag counts and 16 registers to dump the counter values into at the end of each phase bin. During the next phase bin the 16 register values are added one at a time into memory (A-RAM in this block diagram). For each register that needs to be added onto memory, the old value corresponding to that phase bin of that register or lag is retrieved, added to the current register value and stored back in memory. The next register follows, and the cycle repeats until all 16 lags have been added into memory.



sequence of signals necessary for operation of the ramadder boards is generated (§8.4).

The algorithm behind "folding" involves adding new data to the old data in a particular phase bin, a read-add-write (RAW) sequence. The counts from each lag are accumulated into a counter for the duration of a phase bin, then added to the value stored in memory for that lag and phase bin. There are 64×1024 memory words per board, one word for each phase bin of each lag.

Each phase bin lasts anywhere between 1.5 μs and a few ms, after which the contents of the counter are dumped into a register and the counter is quickly cleared before it starts accumulating lag counts for the next phase bin. Now the contents of each register have to be added onto memory before the end of the current phase bin, which means that 64 RAW cycles (one for each register on a board) have to be performed in less than 1.5 μs, the minimum phase bin size. Since integrated circuits can perform may operations in 1.5 μs, the lags can be multiplexed, and a ramadder board is divided in four sections that work in parallel to serially process 16 lags each. Figure 8.2 shows a block diagram for the basic ramadder board section.

Ideally, each section would fold more than 16 lags, but then each RAW cycle would have to be done very quickly, at which point the memory chips and the ALU chip that adds the counter value to the old value from memory need to be fast enough to be prohibitively expensive. Mainly because of board space limitations and memory costs, in the end we decided to have four sections of 16 lags each per board, with a minimum RAW cycle time

of 167 ns. Sixteen lags per section is also a convenient number because the lag data signals are count enable lines and the correlator provides exactly one count clock for every 16 lags. The four sections work in parallel, each one processing the 16 lags serially.

This means that the minimum phase bin time is 16×167 ns=2.7 μs, longer than 1.5 μs. This is only a minor problem, due to which pulsars with periods shorter than 2.7 ms cannot be observed with the full 1024 bins of phase resolution. The chips on the ramadder boards are all mounted on sockets (instead of soldered directly to the board) so that they can be exchanged for faster, pin-compatible parts if the need ever arises. Then the RAW cycle time could be reduced below 167 ns and the minimum phase bin will be correspondingly less than 2.7 μs. The ultimate limit in bin size is described in the caption of figure 8.4 below.

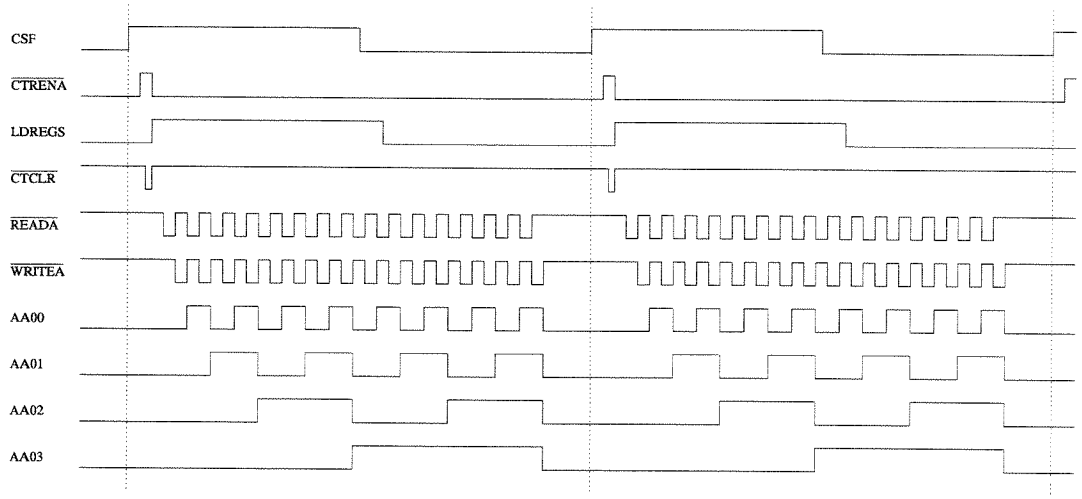
8.3.1 The RAW Cycles

Each section of a ramadder board has to process the data from 16 lags from the previous phase bin before the current phase bin is over. The first thing is to copy the counter values into the registers, which involves the following steps: stopping the 8-bit counters, latching their contents into the 8-bit registers, resetting the counters and freeing them to start counting again. It is necessary to prevent the counters from accumulating lag counts when they are being latched into the registers, to allow their outputs enough time to settle and be stable, otherwise the outputs might be in a transitional state and therefore unpredictable. Thus there'll be a small dead time between phase bins of the order of 80 ns.

When the counters have been latched into the registers and are accumulating lag counts for the current bin, each section has until the end of the phase bin time to do the 16 RAW cycles. This is at least 2.7 μs, as we saw in the previous section. As a matter of fact, the 16 RAW cycles are *always* done in 2.7 μs even if the phase bins are larger than that. From a practical point of view, this is better than having variable clock rates and delays, for then a particular timing combination might fail, and it would be hard to test all the possible configurations ahead of time. It's much better to trigger the beginning of the RAW sequence and then proceed always at the same rate, even if it means having to wait idle for some time until the end of the phase bin. Figure 8.3 summarizes the main events that drive a burst of 16 RAW cycles.

Each RAW cycle occurs in two halves: read/latch and add/write. The 16 registers have a common 8-bit output bus, but their outputs don't interfere because they are all in a high impedance state unless they are being selected. The 16 register outputs are enabled one by one in turn. Each time an output is enabled, it is equivalent to reading the 8-bit register value into the 8-bit bus and making that value available at the adder input. Simultaneously, the previous value for that phase bin and same lag is read from memory. In the second half of the cycle, the 20-bit value from memory is latched into the

Figure 8.3: Control Sequence for a RAW Burst
A burst of 16 RAW cycles is triggered by the beginning of a phase bin, the rising edge of the CSF clock. Two such bursts are shown. For each RAW cycle in the burst, the old value from memory is read, the new register value added to it, and the result is written back to memory. The lag counter is increased by 1 and the cycle repeated. When the 16 cycles are over, the RAW section sits idle until the beginning of the next phase bin. The signal names stand for the following: CSF correlator sampling function, CTRENA counter enable, LDREGS load registers, CTCLR clear counters, READA read A memories, WRITEA write A memories, AA0–AA3 address bits 0–3. Signals that are active-LOW are symbolized with a bar on top.



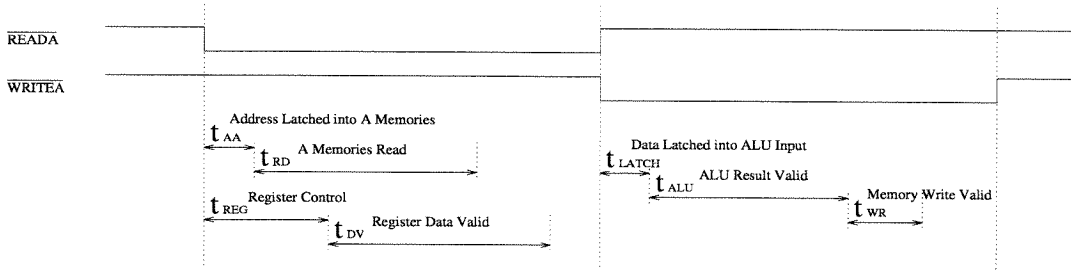
adder input. The latching cannot occur in the first half of the RAW cycle because it takes the memories a certain amount of time to produce stable outputs, but the value *has* to be latched because the memory chips used in the ramadder boards have common input and output data pins. Some time after the 20-bit memory value has been presented to the adder, the result of the 8+20 bit addition becomes valid and stable, and can be written back into memory. Figure 8.4 shows a single RAW cycle in more detail and discusses the reason for the 2.7 μs minimum RAW time.

8.3.2 The Maximum Phase Bin Size

At an input rate of 32 Mbit/s per lag, or 16 Mbit/s after the divide-by-two stage, the 8-bit counters can fill up in as little as 16 μs, which is therefore the maximum phase bin size. This does not mean, however, that pulsars slower than 16 ms cannot be observed, as there is a simple solution to the 8-bit limit. The solution is to divide each phase bin into several sub-bins, each one lasting between 2.7 and 16 μs. Only when the 8-bit counters have been read several times and the desired phase bin time is over will the phase bin counter be incremented. We call this number the number of **passes** through the 8-bit counters. For example, a 24 ms pulsar will be observed with 1024 phase bins of roughly 24 μs each, but each phase bin will be implemented as 2 passes of roughly 12 μs each. It makes little

Figure 8.4: A Single RAW Cycle

A RAW cycle is divided into two halves. During the first half, one of the sixteen 8-bit registers is enabled onto one of the adder inputs, and simultaneously the memories are addressed and read. Starting in the middle of the cycle, the "old" 20-bit value from memory is latched into the other adder input, and the memories are switched from read to write. The addition occurs during the second half of the RAW cycle, and the actual memory write occurs at the very end of the cycle. The period of the RAW cycle can be decreased until we run into conflicts with the time it takes to read the memories, to enable the register outputs, or to perform the addition. The longest of these three times (t_{AA}+t_{RD}, t_{REG}+t_{DV} and t_{LATCH}+t_{ALU}+t_{WR}) determines how long the RAW cycle must be. With the current chip families that we use in the ramadder boards, the ultimate RAW cycle time is 140 ns and is determined by the register output enable sequence, of 70 ns. This corresponds to a minimum phased bin size of 2.24 μs, smaller than the quoted 2.7 μs. The current limitation to 2.7 μs comes from the control board, not the ramadder boards. An asymmetrical RAW cycle could be used to shrink the RAW cycle time even further, but it would require changing the control board significantly.



difference whether we have 2 passes of 12 μs or 3 passes of 8 μs, other than with two passes the counters are only momentarily disabled twice instead of three times, at the beginning of each pass when their values are copied to the registers, so we tend to use as few passes as possible per bin.

8.3.3 The Output Memories

The RAW sequence will proceed forever as long as the integration lasts. Because of the limited real estate for chips on the ramadder boards, we had to restrict the memories to be only 20 bits wide. Therefore, each section has 16×1024×20 bits of memory organized as 16k 20-bit words. This means that the maximum accumulation time per phase bin is 2^{20} bit/16Mbit/s = 0.066 s, and given that there are 1024 bins per period, the longest integration is 67 seconds. For real astronomical data the maximum expected data rate from the correlator will be 20 Mct/s instead of 32 Mct/s (§7.5), so we can reasonably extend the integration time to some 100 seconds.

In any case, the memories need to be read out once every 100 s or so to prevent overflow. The actual integration can continue while they are being read out because there is a second set of memories on each section (the **B memories**) that are used to implement a special form of double buffering. For the sake of clarity, we will refer to the original memories as the **A memories**. Here is how the double buffering scheme works. Suppose that we are

observing a 0.1 s pulsar, and we want to read and clear the A memories once every 100 s, or every 1000 periods. Let M=1000, and imagine a **period counter** that goes from 1 to M (1 to 1000 in this example). Such a period counter does of course exist on the control board and counts from 0 to $M-1$, but the example is simpler if we let it begin at 1 rather than 0.

During periods 1 to $M-1$, the A memories are used for the RAW cycles and the B memories are left completely alone. During period M, the last a group of M periods, the result from the adder is written to both the A and the B memories. At the end of period M, the B memories hold an exact replica of what is in the A memories. Now the control board has $M-1$ periods to read the B memories before it is time to copy the A memories onto them again. Figure 8.2 shows how it is possible to write to both memories at the same time. Reading the B memories can be done completely asynchronously from the RAW cycles and the A memories, as long as the current period is not the Mth of each group.

Now period $M+1$ is essentially the same as period 1. We need to clear the A memories so that we start counting from zero again, but we don't want to spend any time clearing them at the cost of not observing. The solution is to instruct the adder to copy the 8-bit value onto the new 20-bit output instead of adding it to the old 20-bit value from the A memories, which it will now ignore. This is possible because the adders are not just adders, but complete arithmetic logic units that can perform a number of operations. In this case, during periods 1, $M+1$, $2M+1$, $3M+1$ and so on, the ALUs will execute the operation $S=A$ instead of the usual $S=A+B$.

8.3.4 Count Rates in the Ramadder Boards

In this section we summarize everything having to do with data rates in the ramadder boards, purely for reference. The pulsar backend processes 512 lags from the correlator, each one a 32 Mbit/s serial data stream. The lags are processed 64 to one ramadder board, and on each ramadder board there are four sections that process 16 lags apiece. The 32 sections in the pulsar backend (8 boards times 4 sections per board) work in parallel, but the 16 lags they each process are done one after the other, sequentially.

Each lag first goes through a flip-flop that effectively divides the data rate from 32 to 16 Mbit/s. After that, the lag data is accumulated into an 8-bit register for the duration of the current phase bin, from a minimum of 2.7 μs to a maximum of 16 μs. If the pulsar is faster than 2.7 ms, we reduce the number of bins to less than 1024, until the bin size is at least 2.7 μs. If the pulsar is slower than 16 ms, we accumulate several times into the same position in memory, each pass through the 8-bit counters lasting at most 16 μs.

Each memory word is only 20 bits wide and will fill up in 65 ms of continuous integration, but since each phase bin is only addressed once per period, the maximum integration

time is of 65 ms×1024 bins=67 s. While that is true for the very maximum data rate possible, astronomical signals will have smaller data rates, around 20 Mbit/s instead of 32 Mbit/s (§7.5), and we can expect to be safe with integrations of up to 100 seconds.

Each of the eight ramadder boards must be read at least once every 100 seconds, but further accumulation can continue in the computer, where each memory word is 32 bits long. Sinced the ramadder boards produce 2 Mbytes of data every 100 seconds, it is common practice to add together several such 100-second dumps, or else disk space becomes an issue after a few hours of observation. This, of course, can only be done when data points can be separated by as much as a few minutes; in the case if ISS observations one often needs to read the ramadder boards every 10–20 seconds and store and process those observations separately.

8.3.5 Cross Talk and Double Buffering

The ramadder boards are extremely dense from the point of view of a board designer. The data paths are 8 bits wide before the adders and 20 bits wide afterwards. The memory addresses are 14 bits wide (enough to address 16×1024 phase bins), and there are a good number of control signals going to most of the chips on the board. This requires a total of 8 signal layers in the printed circuit board, and the two thousand or so traces per board cross again and again, layer after layer.

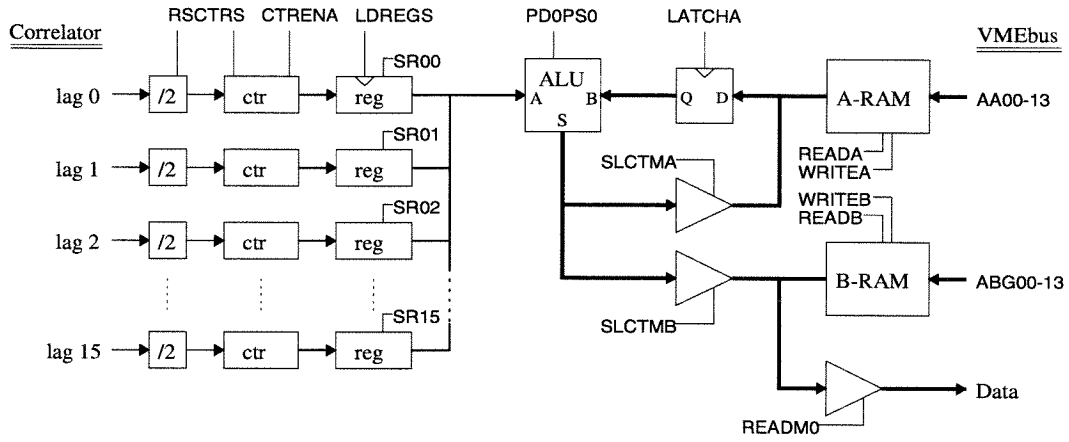
But the real problem is that the boards are very densely packed with chips (209 per board) and there is little space between the pins for so many traces. This leads to forced routing and a small degree of crosstalk between some of the signals. The first printed circuit boards had very severe crosstalk and forced us to improve the layout. The second PCB version has almost no crosstalk.

There is one situation in which crosstalk still matters, though, and that is between the B memories and the rest of the board. The crosstalk would not exist if the B memories were read synchronously with the rest of the signals on the board because then a small amount of crosstalk can be tolerated (all the signals change levels at the same time; it just takes them a little longer to settle). However, since we read the B memories asynchronously with respect to the rest of the board, it induces small spikes on signals that are supposed to be stable, and those spikes are sometimes interpreted incorrectly by the remaining circuitry.

Unfortunately, this means that the double buffering scheme cannot be used to read out the B memories while the A memories are integrating, as reading the B memories affects the data being written into the A memories. The "solution" is to alternate long and short integrations. The long integrations are used to collect data and to copy the A memories to the B memories, which happens synchronously. Then a small integration is done during which the B memories are read out by the computer, corrupting the A memories. Although we still copy the A memories to the B memories at the end of the short integration, we

Figure 8.5: The Signals in a Ramadder Board Section

This figure is the same as figure 8.2, but contains the main signals that control the operation of the section. The signals are described in table 8.1 and the text. The 16 dividers, counters and registers share the signals `RSCTRS`, `CTRENA` and `LDREGS`, while each 8-bit register gets a different register select signal `SRi`. The addresses for the B memories are called `ABG` because they are *gated*, and are equal to the A memory addresses when written to, and equal to the VMEbus address when read.



now avoid reading the B memories in the next long integration, thus not corrupting the A memories.

It's a pity that so much effort was put into the double buffering scheme, because we could have used the board space taken by the B memories.

8.4 The Control Board

The control board generates 60 control signals for the ramadder boards. Each one of the eight ramadder boards uses all 60 signals to function, in parallel with the other ramadder boards. The control signals are distributed from the control board via rows A and C in the VMEbus P2 backplane. They are briefly described in table 8.1. Figure 8.5 shows the same ramadder section as in figure 8.2 above, but this time with the control signals.

In addition to keeping the ramadder boards running during an observation, the control board must keep track of time so that a correct pulse arrival time can be calculated later on. For this purpose, it uses two reference signals from the observatory: the 1 second clock and the 5 MHz frequency standard. The 1 s clock is used to start each observation at a well-known time; the 5 MHz reference is used to keep track of time between ticks from the 1 second clock, and to generate the pulsar spin frequency for folding the correlator data (also called the "pulse repetition frequency").

Table 8.1: Control Signals for the Ramadder Boards

The control board generates 60 control signals for the ramadder boards to function properly. Of these, 3 control the operation of the 8-bit counters, 17 control the 8-bit registers, 1 controls the ALUs, 18 control the A memories and 21 control the B memories. All four sections on each ramadder board function in parallel. The only difference between them is the `READBi` signal, since the B memories are read out section by section, board by board. The control signals are distributed through rows A and C of the P2 VMEbus backplane (§A).

Signal	Description
8-bit counters:	
`CTRENA-B`	Enable the 8-bit counters
`RSCTRS`	Clear the 8-bit counters
8-bit registers:	
`LDREGS`	Load the 8-bit registers
`SR00-15`	Select one of sixteen 8-bit register outputs
ALUs:	
`PD0PS0`	Execute "$S=A$" instead of "$S=A+B$"
A memories:	
`READA`	Read the A memories
`LATCHA`	Latch A memory outputs into ALU inputs
`SLCTMA`	Buffer ALU outputs into A memory inputs
`WRITEA`	Write data into A memories
`AA00-13`	14-bit address for the A memories
B memories:	
`SLCTMB`	Buffer ALU outputs into B memory inputs
`WRITEB`	Write data into B memories
`READB`	Read B memories
`READM0`	from section 0
`READM1`	from section 1
`READM2`	from section 2
`READM3`	from section 3
`ABG00-13`	14-bit address for the B memories

8.4.1 Generating the Pulsar Spin Frequency

In order to fold the correlator data synchronously with the topocentric pulsar period, the control board must generate a signal with the same frequency as the pulsar's spin. It actually generates a frequency equal to the pulsar spin frequency *times* the number of pulse phase bins, or $f_{psr} \times N_{bins}$. This signal is called the **correlator sampling function** (CSF), as it determines exactly when the correlator is sampled, once for each phase bin, or rather when the 8-bit counters that accumulate correlator data are sampled into the 8-bit registers.

The CSF signal is produced from a 32 MHz reference by direct digital synthesis (DDS) with a numerically controlled oscillator (NCO). The 32 MHz clock used by the NCO is locked to the 5 MHz reference from the observatory, and is divided in the NCO chip to produce the exact frequency of the pulsar. In the current setup, the frequency resolution of the 32-bit NCO chip is 32 MHz/2^{32}/4=1.9 mHz. This resolution is sufficient, given that the frequency the control board needs to generate is $f_{CSF}=f_{psr} \times N_{bins}$, which is of the order of 100 kHz and therefore much larger than the pulsar spin frequency.

The NCO chip has a 32-bit phase accumulator and a 32-bit Δ-phase register. On each 32 MHz clock cycle, the contents Δ-phase register are added to the phase accumulator, which wraps around through zero when it overflows (i.e., it counts modulo 2^{32}). The CSF signal is obtained directly from the most significant bit in the 32-bit phase accumulator, by division by four, and is therefore a digital signal with a duty cycle very close to 50%.

The topocentric pulsar period changes continuously during the observation and must be updated often in the FPTM. The control board updates the CSF frequency several times a second with the pulsar period predicted from the pulsar's timing model, which is often enough to prevent smearing due to Earth's rotation and to orbital accelerations in binary pulsars. The frequency update is done by overwriting the contents of the Δ-phase register in the NCO chip, which is done in a phase synchronous manner.

8.4.2 The Start of an Observation

All observations done with the FPTM start on a UTC 10-second boundary. This straightforward rule ensures that the start time is always well determined and makes it likely that timing errors will be more easily discovered. Since most observatories do not provide a standard 10 second tick, it is derived in the control board from the 1 second observatory reference. Note that although the pulsar frequency is reproduced to great accuracy with the NCO chip, the rotational phase of the pulsar is ignored; all observations start on a 10 second boundary regardless of the pulsar's phase, and as a result the main pulse will appear at a different phase in every folded profile.

The observing sequence involves several steps: verifying that the control board time is correct, starting the correlator, testing the FPTM, opening all output files, arming the

control board, and waiting for the next 10-sec tick. Only when the 10-sec tick arrives does the actual observation start and the correlator data is folded into the ramadder board memories.

8.4.3 Intrinsic Timing Accuracy

In general, the 1 second clock and the 5 MHz reference are not perfectly synchronous, as one is a time standard and the other is a frequency standard, and as such they have different requirements. This means that a 1 second tick can arrive at any phase of the 5 MHz clock, that is, that the phase difference between the two can be as much as 200 ns. This phase difference is not important in itself but can have important implications.

The basic frequency reference in the FPTM is a 32 MHz clock that is phase locked to the 5 MHz observatory reference. As above, this means that a 1 second tick can have any phase with respect to the 32 MHz clock. Since all the signals that control the ramadder boards are generated from the 32 MHz clock, and not from the 1 second tick, it follows that the beginning of an observation will be unknown by up to 30 ns, the period of the 32 MHz clock. In other words, even though the 1 second tick might be exact, the observation does not start until the next period of the 32 MHz clock, which can happen anywhere within the next 30 ns. This determines the intrinsic time resolution of the FPTM, and should produce timing residuals as large as 30 ns. It is the timing noise floor of the FPTM.

8.5 Clock and Data Issues

To some extent, the ramadder boards and the JPL accumulators perform the same function of accumulation of the lag data from the correlator chips. They both take a number of correlator lag data outputs and a few count clocks. The lag outputs are essentially "count enable" signals, to be sampled at the rising edges of the count clocks. The correlator chips provide four different count clock outputs each, but only one from every two chips is converted from ECL to TTL, and is shared by the JPL accumulators and the ramadder boards. The four clocks per correlator chip are copies of the same basic 32 MHz clock, delayed in phase by multiples of 8 ns. The only phase that works for the ramadder boards is the PHASE2 clock, which is precisely the only one that does *not* work with the JPL accumulator chips.

As a result, the data from the JPL accumulators in the FPTM will be corrupted, especially for the second set of negative lags (lags 48-63 on the board, from correlator chips CC7 and CC8), which have more delicate timing. Data taken with the JPL accumulators will be generally correct, but cannot be trusted even when everything *else* is working perfectly. This is a problem that has no apparent solution unless we add more ECL-to-TTL converter chips on the correlator boards, but is not fundamental because the data

from JPL accumulators is only used in some tests.

Another, more difficult problem, but one which we have solved, is as follows. The correlator outputs run at 32 MHz each. Timing problems do occur at these frequencies, not because of having the wrong trace or cable lengths, but generally because of unequal timing delays through integrated circuits working in parallel. The rather loose timing requirements of a single 32 MHz signal become extremely delicate when 512 such requirements must be satisfied simultaneously. The problem arises because the correlator outputs from different sections on the correlator boards are asynchronous and have count clocks and data that are not time aligned. On the other hand, the ramadder boards are synchronous and expect all inputs to be time aligned. This has forced us to align all the correlator signals, which we do by issuing a multiply reset pulse at the beginning of an observation, and to lock the 32 MHz ramadder clock to the correlator boards. All in all, this is equivalent to phase locking some 545 signals together.

Solving this problem was done in two stages. First we realized that the correlator clocks had to be time aligned, or else different groups of lags would behave differently depending on the relative phase of their count clock with the ramadder clock. Some groups of lags would do very well and some others very badly. Once we started issuing the multiply resets, all 512 lags behaved identically well or identically poorly. The 32 MHz clocks from the correlator and the 32 MHz clock in the ramadder are only linked through the 512 MHz frequency generator, and therefore can have almost any random phase with respect to each other upon power up. The final solution was to phase lock the two groups of 32 MHz clocks to the particular phase relation for which all 512 lags would behave identically perfectly, and this happened when we delayed the ramadder clock by 4.3 m worth of cable (± 1 ft) with respect to the grand correlator clock.

Appendix A

The VMEbus

Abstract

The VMEbus is an industry standard for computers and digital boards that reside in the same crate. It specifies the physical dimensions of the cage, the backplanes and connectors, as well as signals and their electrical properties and timing. This appendix describes the VMEbus standard and a simple VMEbus slave interface designed and built for the Caltech Fast Pulsar Timing Machine.

A.1 Overview

The VMEbus is an industry standard for data exchange between boards in the same crate. It specifies a backplane and a protocol for using it. The backplane specifications not only include physical dimensions, power supplies, slot separation, connectors and traces, but also the electrical properties of anything that connects to the backplane, the proper ways to request the bus, to respond to a request and to transfer data, and the timing of all electrical signals that have to do with the bus. This appendix summarizes the most useful information from the actual VMEbus Specification Manual (Motorola 1985).

A.1.1 VMEbus Formats

The VMEbus comes in two sizes. The small VMEbus has a single backplane containing a 24-bit address bus, a 16-bit data bus and a series of control signals. The extended VMEbus has a second backplane with an additional 8 bits of address and 16 bits of data[1]. The extended VMEbus thus permits 32-bit data transfers over a 32-bit address space and is therefore faster and more versatile. In the FPTM, the correlator crate is A24:D16, while the ramadder crate is A32:D32.

[1]There are actually 43 control signals on the VMEbus, besides the 24 address bits and 16 data bits. The extended VMEbus has the same 43 control signals, plus 32 address bits and 32 data bits.

Table A.1: VME Board Dimensions

Single- and double-height boards have standard height and depth, while triple-height boards have a variety of depths. The thickness of all VME boards must be less than 1.6 mm, the only limitation being the guide rail width. All dimensions in the table are in mm.

Format	Height	Depth
3U	100	160
6U	233.4	160, 218
9U	366.7	277, 400

The pin layout of the J1/P1 and J2/P2 connectors[2] is shown in Tables A.2 and A.3. In triple-height crates there is sometimes a P3 backplane underneath P2, but there are no VMEbus specifications attached to it. For this reason one is free to use all 96 pins in each P3 connector for special purposes, in addition to the 64 unused pins from the P2 backplane.

Single height VME crates are called **3U**, double height **6U**, and triple height **9U**. A24:D16 systems are normally 3U or 6U, while A32:D32 systems must be either 6U or 9U, since they must use both the P1 and the P2 backplanes. Table A.1 shows the standard heights and depths for the different VME board sizes. There is an absolute maximum of 21 boards per cage, equivalent to a rack width of 50 cm.

A.1.2 Termination

Most control signals are active-low, and terminators on each end of the backplane ensure that the undriven signals remain high. For address and data signals, the termination is not absolutely necessary, since the board driving these lines will drive them low or high as necessary. However, having the terminators gives the lines the proper impedance and reduces reflections and bouncing. For control signals, on the other hand, the terminators are always necessary. Signals like interrupt or bus requests could cause havoc if they were left floating.

Each signal is terminated with a 330 Ω resistor to +5 Volt and a 470 Ω resistor to GND, at each end of the backplane. The resulting undriven level is +2.94 V, which is TTL logic HIGH. Similar termination can be achieved with 220Ω/330Ω resistor pairs.

In the FPTM we use some of the undefined pins in the P2 and P3 backplanes. In order to use them we had to fabricate new backplanes that would have traces connecting those pins from slot to slot, from end to end of the backplanes. In addition, we had to build new terminator printed circuit boards that would not only terminate the standard VME

[2] P1 and P2 normally refer to the backplanes, while J1 and J2 refer to the connectors that plug into the backplanes. P1 is always the top backplane.

Table A.2: VMEbus P1 Layout

Signal layout in the VMEbus P1 backplane. The signals that are active-low are marked with an asterisk. All control signals, address bits and data bits are high when undriven, thanks to terminators on each end of the backplane. The power pin "+5 V STDBY" is sometimes used in crates that have a backup power supply and provides alternative power to volatile memory in case of main power failure.

Pin	Row A	Row B	Row C	Pin
1	D00	BBSY*	D08	1
2	D01	BCLR*	D09	2
3	D02	ACFAIL*	D10	3
4	D03	BG0IN*	D11	4
5	D04	BG0OUT*	D12	5
6	D05	BG1IN*	D13	6
7	D06	BG1OUT*	D14	7
8	D07	BG2IN*	D15	8
9	GND	BG2OUT*	GND	9
10	SYSCLK	BG3IN*	SYSFAIL*	10
11	GND	BG3OUT*	BERR*	11
12	DS1*	BR0*	SYSRESET*	12
13	DS0*	BR1*	LWORD*	13
14	WRITE*	BR2*	AM5	14
15	GND	BR3*	A23	15
16	DTACK*	AM0	A22	16
17	GND	AM1	A21	17
18	AS*	AM2	A20	18
19	GND	AM3	A19	19
20	IACK*	GND	A18	20
21	IACKIN*	SERCLK	A17	21
22	IACKOUT*	SERDAT*	A16	22
23	AM4	GND	A15	23
24	A07	IRQ7*	A14	24
25	A06	IRQ6*	A13	25
26	A05	IRQ5*	A12	26
27	A04	IRQ4*	A11	27
28	A03	IRQ3*	A10	28
29	A02	IRQ2*	A09	29
30	A01	IRQ1*	A08	30
31	−12V	+5V STDBY	+12V	31
32	+5V	+5V	+5V	32

Table A.3: VMEbus P2 Layout

All control signals are in the P1 backplane. The P2 backplane has the address and data pins for the A32:D32 extension, as well as additional power and ground pins for the larger boards. Rows A and C are not used in the extension and are free for the user to define and use. For the same reason, however, the pins in rows A and C not only are not terminated in commercial backplanes, but they are not connected from slot to slot through the backplane.

Pin	Row A	Row B	Row C	Pin
1	–	+5V	–	1
2	–	GND	–	2
3	–	RESERVED	–	3
4	–	A24	–	4
5	–	A25	–	5
6	–	A26	–	6
7	–	A27	–	7
8	–	A28	–	8
9	–	A29	–	9
10	–	A30	–	10
11	–	A31	–	11
12	–	GND	–	12
13	–	+5V	–	13
14	–	D16	–	14
15	–	D17	–	15
16	–	D18	–	16
17	–	D19	–	17
18	–	D20	–	18
19	–	D21	–	19
20	–	D22	–	20
21	–	D23	–	21
22	–	GND	–	22
23	–	D24	–	23
24	–	D25	–	24
25	–	D26	–	25
26	–	D27	–	26
27	–	D28	–	27
28	–	D29	–	28
29	–	D30	–	29
30	–	D31	–	30
31	–	GND	–	31
32	–	+5V	–	32

signals, but our signals as well. We used the same exact termination as for VME signals, and all the new signals are active-low.

A.1.3 Master and Slave Interfaces

A VMEbus read or write cycle is always initiated by a **bus master**. The board that responds to a read or write request is called a **bus slave**. In general, there is just one master per VME cage, it resides in slot 1 (the leftmost slot) and it is the computer where the control programs run. The rest of the slots are normally filled with slave boards that respond to requests from the computer, or master.

There are two ways for the other boards to initiate a bus cycle. They can become a bus master, or they can issue an interrupt request and wait for the computer to initiate a bus cycle in response to the interrupt request. Becoming a bus master gives the requesting board total control of the bus, but it also requires a more sophisticated hardware interface. Issuing interrupts is relatively simple, which makes a simple slave interface with interrupt capabilities relatively easy to implement. Such a slave interface will have most of the functionality needed in standard applications.

There exist commercial VMEbus interface chips for user developed boards. These usually consist of one or two large chips (PGA packages, 81–149 pins) and their main function is to make the VMEbus transparent to a Motorola 68k bus. They are very convenient if the board being built is controlled by a 68k-type microprocessor, but if not, they are of no help whatsoever. Not only they are expensive and difficult to use, but they take a lot of real estate on the board and they have more power than is needed.

As a result we have developed our own VMEbus interface. It performs the minimal functions of a VMEbus slave, and can issue interrupts. All in all it fits in four small programmable chips and is cheap and easy to use.

A.1.4 Read and Write Cycles

All VMEbus read and write cycles are initiated by the bus master, usually a computer. The master first requests the bus and then initiates the bus cycle. "Requesting the bus" means nothing other than waiting until the bus is idle and then grabbing it. "Initiating the cycle" means putting an address on the address bus, driving the address strobes low, then waiting for a response from the slave board that is being addressed.

In the case of a write cycle, the master puts data on the data bus as well, and drives the data strobes. In a read cycle, it expects the responding slave to drive the data lines, but the master drives the data strobes itself because they convey information to the slave, such as the data bus width.

The terminology of read and write cycles is always from the point of view of the bus master. Thus, a write cycle always implies that the master drives both the address and

the data buses, while a read cycle implies that the master drives the address bus but reads data from the slave, who must drive the data bus.

In either case, the master must wait until the slave acknowledges the transaction. This acknowledgement is given after the slave has put the data on the bus (in the case of a read cycle), or when it has read the data from the bus (in a write cycle). If an acknowledgement does not come from any of the boards on the bus, a timeout will occur and the master will generate a bus error. This normally results in a program aborting in the master, or a software trap being executed if one has been set.

A.1.5 Interrupts

Issuing an interrupt allows an otherwise passive board to ask for attention from the bus master. The bus master, normally a computer, responds to the interrupt request and performs the necessary actions. The actions depend on the application, but in general they involve reading data from the slave that issued the interrupt. Peripherals like keyboards issue an interrupt for every keystroke, in which case the computer stops what it is doing, reads and processes the character from the keyboard, and then returns to what it was doing.

Each interrupt is characterized by two numbers: the **interrupt level** and the **interrupt vector number**. The interrupt level (a number between 0 and 7) establishes the urgency of the interrupt with respect to other interrupts. A low level interrupt will be serviced only when no interrupts of equal or higher level are pending. On the other hand, a high level interrupt will preempt a low level interrupt. Only when the higher level interrupt has been serviced will the lower level interrupt continue to be serviced.

The interrupt vector number (a number between 0 and 255) tells the computer which software subroutine to execute in order to service the interrupt. The service routine for a keyboard interrupt might just read a character code from the keyboard, then exit. There can be one or more service routines linked to a vector number, or none at all. If there are no entries for a vector number in the **interrupt vector table** (each entry being a pointer to a service routine), then no action is taken. But at the same time, no error is generated since the only effect of the dummy interrupt was a small waste of time.

If several service routines are attached to a vector number, then each one of them is polled in turn until one of them acknowledges the interrupt. If none of the routines claim it, then no action is taken and that interrupt is declared serviced.

A.2 The Technical VMEbus

In this section we treat some technical aspects of the VMEbus that would have clouded the overview. First of all, we consider the actual control signals that play a role in data

Table A.4: Address Modifier Bits

The address modifier bits in the VMEbus indicate what the address width is (16, 24 or 32 bits) and the type of transfer: whether supervisory or non-privileged, and whether block, program or data. This table list only 18 of the 64 possible address modifier codes; the remaining 46 are reserved and slaves should not respond to them.

Bits	Description
5-3	101 = short or 16-bit address
	111 = standard or 24-bit address
	001 = extended or 32-bit address
2	1 = supervisory
	0 = non-privileged
1-0	11 = block transfer
	10 = program transfer
	01 = data transfer

cycles and interrupt requests. And second, we elaborate on a few issues that would have helped us when we first started to play with the VMEbus.

A.2.1 Read and Write Cycles Revisited

A data cycle starts when the bus master asserts the address strobe AS*, which tells all slaves that an address has been put on the address bus. Whether this address is 16, 24 or 32 bits wide depends on the address modifier bits AM5-0, described in table A.4, and whether the data should be 8, 16 or 32 bits wide is determined by LWORD*, DS0* and DS1* (see table A.5). In some cases two bus cycles are required for a single transfer, and then the two cycles are executed in a manner completely transparent to the software. One such case is a 32-bit data transfer over a 16-bit backplane.

It must be noted that the VMEbus address bus lacks bit A00, that is, it has only bits A23-A01 in the standard backplane and A31-A01 in the extended, A32:D32 backplane. This does not mean that single-byte transfers are impossible, as long as another control signal is used, in this case DS1* (or rather a combination of DS1* and DS0*).

All the slaves on the bus decode the address and address modifier bits when AS* is asserted, and the one that is addressed will get ready to either read a word from the data bus (write cycle) or write one (read cycle), depending on WRITE*. When the master asserts the data strobes (one or both, generally known as DS*) the slave will then read or write the data, and then assert the data acknowledge signal DTACK*. This signals the bus master that the transfer is over and that it can release the address bits, the address modifier AM5-AM0 and the address and data strobes, as well as LWORD* and WRITE*.

Table A.5: Data Transfers Over the VMEbus

The VMEbus has specifications for single-byte, double-byte and quad-byte transfers. On a single-backplane 16-bit data bus, quad-byte or 32-bit transfers are not possible in a single bus cycle. In that case, two cycles are automatically executed by the VMEbus interface hardware in a perfectly transparent way to the software. Since memory is organized in 32-bit words, byte[0] refers to the most significant byte and byte[3] to the least significant byte. The most significant byte has the smallest address.

Transfer	D31-D24	D23-D16	D15-D08	D07-D00	DS1*	DS0*	A01*	LWORD*
byte[0]			byte[0]		0	1	0	1
byte[1]				byte[1]	1	0	0	1
byte[2]			byte[2]		0	1	1	1
byte[3]				byte[3]	1	0	1	1
byte[0–1]			byte[0]	byte[1]	0	0	0	1
byte[2–3]			byte[2]	byte[3]	0	0	1	1
byte[0–3]	byte[0]	byte[1]	byte[2]	byte[3]	0	0	0	0

A.2.2 Interrupts Revisited

At the hardware level, any board on the bus can issue an interrupt by pulling one of the interrupt request lines low. The request will be caught by a bus master, who will respond by initiating the so-called **interrupt acknowledge daisy-chain**. The interrupt acknowledge daisy-chain goes through all the slots on the cage. Each slot has an IACKIN* signal and an IACKOUT* signal. If the board on that slot receives IACKIN*, and is the one who originally issued the interrupt request, then this board must stop the daisy-chain and repond to the master's query by putting an 8-bit interrupt vector number on the data bus. If the board is not responsible for the request, then it must pass the master's query to the next board by driving IACKOUT*.

Most of the signals on the VMEbus are connected from slot to slot all the way across the backplane. All the boards then have access to the same signal and any one of them can drive it with identical priority. The interrupt daisy-chain is different because the boards closest to slot 1 will receive the master's response or query to an interrupt request before the other boards. IACKIN* and IACKOUT* are therefore not connected all the way through, but rather one slot's IACKOUT* is connected to the next slot's IACKIN*. If a slot does not have a board plugged in it, it must instead have a jumper shorting IACKOUT* to IACKIN* in order to preserve the daisy-chain. If the jumper is missing, a bus error could result when the bus master responds to an interrupt request but the response cannot make it to the requester, who then cannot clear the interrupt request before a bus timeout occurs.

There can be several interrupt requests at the same time as long as they are of different level. When the bus master responds to one of the requests, it will encode the interrupt vector level into the address bus, as bits A03-A01. This is how a slave board can check what

level interrupt the master is responding to, and can decide whether to stop the daisy-chain or pass it on to the next board.

Once a slave intercepts the daisy-chain, places the interrupt vector number in D07-D00 and asserts DTACK*, the hardware part of the interrupt protocol is over. At this point it is all left up to the bus master, who now has an interrupt vector number and should execute its interrupt service routine as soon as possible. Nothing else will happen, however, if there is no service routine attached to that vector or the service routine does nothing.

A potential problem can occur when the slave board cannot drive the data bus properly, for then it can effectively place the wrong interrupt vector number on the bus and thus activate the wrong service routine. Some service routines are dangerous and can trigger a system reboot.

A.3 The FPTM VMEbus Interface

We have developed a VMEbus slave interface for the boards in the FPTM. It has the capability to respond to read and write cycles on the VMEbus and to generate interrupts. From the board side the interface is very simple and has but three control signals: a data strobe DS*, a write bit WR*, and an interrupt request IRQ. DS* and WR* are output to the board, while IRQ is an input from the board.

The interface continuously monitors the VMEbus. When the proper address is strobed the board is selected, and some of the address bits are decoded and demultiplexed into a function for the board (such as reset, read status register, take data or load a particular counter with data from the bus). Some of the functions will read data from the bus, some will write data, and some (like reset) will ignore the data bus.

A few VME clock cycles after the board was selected, the interface terminates the transaction by asserting DTACK*, and immediately becomes ready for another selection.

The interface is simple because the board can act only as bus slave, and thus never has to request and capture the bus nor drive all the control signals. The capability of generating interrupts is also simple to implement, but extremely important because it allows the board to start a transfer, even though it's a slave, by asking the bus master to do it.

In order to generate an interrupt, the board has to drive the IRQ line high. The interface takes care of the hardware protocol and gives the VMEbus master an interrupt vector number. The vector number is burned into the interface for that board, just as the bus address to which the interface will respond.

In addition to the four programmable chips, a successful interface will also include a set of drivers for the address and data lines. The drivers fulfill two purposes: first, to make sure that the master drives no more than one gate-equivalent per control or data signal per

board; and two, to make sure that the board only drives the data bus during read cycles, and that when it does it does so with sufficient drive capability. Bi-directional drivers are recommended for this function, and they can easily be controlled by the interface with the `WR*` signal.

Appendix B

The PIO Board

Abstract

The word **PIO** stands for **Parallel Input and Output**. The PIO board is a VME board that can perform simultaneous input and output of digital parallel data via a ribbon cable. The most general use of the PIO boards is for data exchange between two computers in different VME cages. One PIO board is required on each crate, and the two computers exchange data via the PIO boards.

B.1 Introduction

The PIO board is a VME board designed for input and output of digital data. There are separate input and output data buses, each carrying an independent strobe line, which allow simultaneous input and output. In addition, the input bus has a small buffer that frees the local computer from reading each data word as they arrive.

The buffer is a First-In-First-Out (FIFO) memory, and as such has 3 useful flags: the empty flag `EF`, the half-full flag `HF`, and the full flag `FF`. These 3 bits of information are made available as part of the PIO board status register.

B.2 Receiving Data

In order to understand how the PIO board works, consider one that has just been reset by the computer. The input buffer is empty, and therefore the empty flag `EF` is true, while the other two flags, `HF` and `FF`, are false; the input buffer is empty and neither half-full nor completely full. When the first data word arrives (is strobed) through the ribbon cable, it is automatically written into the buffer. Thus the empty flag `EF` becomes false, and all 3 flags now read false.

The transition of `EF` from true to false generates a VMEbus interrupt that tells the computer that the input buffer of the PIO board contains data that may now be read. If, for some reason, one chooses to ignore the interrupts, then it becomes necessary to

continuously poll (query) the PIO board status register until EF is false, but such continuous monitoring is wasteful.

At this point the computer should read a word, then check whether the buffer is empty or still contains data (EF still false). If it does still contain data, another word must be read, and so on. Two things are helpful at this point. First, if HF is true, then the computer can read at least N/2 words without checking EF, N being the size of the input buffer in words. And second, once EF is true again, the computer can stop checking the PIO board since the next incoming word will generate an interrupt anyway.

B.3 Sending Data

There is no output buffer on the PIO boards. However, the status flags of the remote board are available through the ribbon cable, and therefore one can view the input buffer of the remote board as the output buffer of the local board. Of course, this is only true when another PIO board is at the remote end.

Before writing a word out, the computer must check whether the remote buffer is full. If the remote FF is true, then it must wait and keep polling. No interrupt is generated when FF becomes false, but the time spent waiting will be small because the remote computer is supposed to read its data right away.

Once FF becomes false, the computer can write at least one word. As when reading, the remote HF should be checked, for if it is false then N/2 words can be written at once without having to check FF. On the other hand, if the remote HF is true, then words must be written one at a time, and FF must be checked to prevent buffer overruns.

When the PIO board sends a word on the ribbon cable, it actually places the bit values on the cable and then sends a strobe that writes that word into the input buffer of the receiving PIO board.

B.4 Addressing the PIO Board

The PIO board is accessible to the computer through the VMEbus. Simple read and write operations are sufficient, since the PIO board is placed within the memory space of the computer. Four functions can be addressed: reset the board, read the status register, read a data word from the input buffer, and write a word out to a remote PIO board. The reset function is the only one that does not carry an argument, and can therefore be issued either in a write or read cycle from the computer.

The input and output words, as well as the status register, are 8 bits wide. Thus communication with the board is byte-by-byte. Table B.1 shows how to address the PIO board.

Table B.1: PIO Board Addressing

Meaning of the VMEbus address bits when accessing the PIO board. The PIO board is effectively an A24:D08 board, hence the 24-bit addresses. Of the 24 bits, the highest byte selects the board itself. The next 5 bits select one of four possible functions, while the data byte is transmitted as the lowest byte of the data bus, D07-D00 (if A01 where 0 instead of 1, the data byte would be placed in D15-D08 – see also table A.5).

Bits	Description
A23-A16	select the board. For example, 0xE5
A15-A11	select the function
	00000 = reset the board
	01001 = read one byte of data
	00010 = write one byte of data
	01011 = read status register
A10-A02	not decoded in PIO board, use zeroes or ones
A01	must be equal to 1
A00	not present in the VMEbus

B.5 The Status Register

The **status register** is an 8-bit register that gives not only information on the local FIFO status, but also about the remote FIFO status should another PIO board be installed at the end of the ribbon cable. Table B.2 describes the status byte.

The two unused bits will normally read high, and if a remote PIO board is not connected, the three remote flags S6-S4 will be undefined and might read high or low.

Table B.2: PIO Board Status Byte

Breakdown of the VMEbus status byte. Bits S2-S0 contain the three flags from the local FIFO. When a remote PIO board is installed, bits S6-S4 are the remote FIFO flags.

Bits	Description
S7	Unused
S6	Remote `FF*`. 0 = output buffer full
S5	Remote `HF*`. 0 = output buffer half full
S4	Remote `EF*`. 0 = output buffer empty
S3	Unused
S2	Local `FF*`. 0 = input buffer full
S1	Local `HF*`. 0 = input buffer half full
S0	Local `EF*`. 0 = input buffer empty

B.6 History and Particulars

The FPTM has two VME computers, one in the correlator crate and the other in the ramadder crate. The correlator computer is ROM-based, that is, does not have a hard disk, and can only execute modules that reside on a ROM, or that it loads from the outside. The "outside" normally means the ramadder computer, since it also runs OS9 and has a hard disk.

Originally, the two VME computers in the FPTM communicated through a serial port, which at 9600 baud was too slow a link. Unfortunately, at higher baud rates the connection is unreliable because the serial ports are asynchronous. It seemed easiest to build a small board that would handle parallel communications between the two boards, and thus the PIO boards were born. Other alternatives were considered, such as an ethernet board for the correlator computer, but they were expensive and not always feasible because of the small size of the ROM.

The first version of the PIO board was wire-wrapped and had two 16-bit single-ended buses. Because of the length of ribbon cable, noise became an issue and a problem, and the printed circuit board version of the PIO board was designed with two differential 8-bit buses instead of the two 16-bit original buses.

The current version is still the first printed circuit board. It has two 8-bit buses, one for output and one for input, and the FIFO is 512 bytes deep (N=512). From the computer point of view, the board does only 8-bit transfers, so the computer must perform four transfers per 32-bit word. It is always good to do error checking after transferring data to ensure that no data have been corrupted.

The current PCB version contains a few bugs that can be fixed with wire and a soldering iron. Some of them can even be fixed from the software, as indicated below.

1. The 12 outgoing bits (8 data bits plus, 3 FIFO status bits and the write strobe) are inverted. It is absolutely necessary to fix the polarity of the write strobe, since it is its HIGH-to-LOW transition that latches the outgoing byte into the remote FIFO. Correcting the write strobe involves cutting the two traces from the differential line transmitters and making new (inverted) connections with thin wire.

   The 11 data bits (8 actual data bits plus 3 FIFO status bits) are easier to invert in software than in hardware. The status bits may be inverted after being read, while the 8 data bits can be inverted either when being sent or when being received.

2. The input write strobe should be terminated so that the strobe bit is always false unless asserted by the remote board. Otherwise, when the remote board is not connected, the differential line receivers will have a definite state, and not oscillate at random, creating ghost writes and as a result ghost interrupts.

Proper termination for the input write strobe is 20 kΩ to +5 V on the positive input `WRIN+` and 20 kΩ to GND on the inverted input `WRIN-`.

3. Ideally, the differential line receiver inputs for bits S6-S4 should also be terminated, but with the positive input to GND and the inverted input to +5 V. This way if the remote board is not connected, all three remote flags will read true, which is an impossible configuration, and the software will know that the remote board is missing.

Appendix C

The Digital Interface

Abstract

The Digital Interface is a set of boards that allows a VMEbus computer to control remote circuits or modules. It comprises a PIO board in the computer cage, an intermediate fan-out box, and a small interface board located inside each of the remote modules to be controlled.

C.1 Introduction

Remote observing requires that the computer in charge of the FPTM be able to control every module of the FPTM. The computer should be able to set the local oscillator frequencies, the observing bandwidth, attenuation and clock rates, and it should be able to reset every module that potentially needs resetting. Since the computer has only direct access to the modules in its backplane, we designed and built an interface for the computer to have control of all the remote modules.

We already had a rudimentary interface to the outside world, namely the PIO board: it gives the VMEbus computer access to a ribbon cable carrying parallel digital data. The new digital interface is based on the PIO board, and for simplicity we decided that information would only flow outwards, that is, the computer should be able to write data to remote modules, but not read data from them.

Since the PIO board has a single ribbon cable, we created a fan-out box (also called **Middle Board**) that connects to several **End Boards**. In this manner the computer, through the PIO board and the middle board, can access a series of end boards, each of which resides inside an instrument or module that we want to control. Each end board contains two 32-bit registers that can be loaded from the computer, as well as two strobe signals. Once connected, the two 32-bit registers determine how the remote instrument or module will behave.

C.2 Data Flow Overview

The digital interface hardware consists of one PIO board, one middle board, and several end boards. Normally each end board is located inside a FPTM subsystem, and can control one or more modules inside the subsystem. How many it can control is determined by the 64 bits of data and the 2 strobes. In the FPTM each end board normally controls two instrument modules (say one module for each of two polarizations), and each module requires less than 32 data bits.

Controlling a board means sending a digital number to it. Some modules are asynchronous and their behavior will change as soon as its 32-bit register is updated. Other modules are synchronous and will load the 32-bit value only when they are directly strobed. In order to support both types of modules, the 32-bit registers are latched (they retain the 32-bit value until addressed again by the computer), and there are two strobe signals in each end board for the remote modules.

When setting the configuration of the FPTM, the control computer will send a series of control words to all the modules it controls. Each control word can be as large as 32 bits, and for simplicity we always send 32 bits whether they are necessary or not. Setting the 32 bits of a 32-bit end board register requires several write cycles from the computer, since each write cycle is limited to 8 bits by the PIO board.

The actual protocol for data transfer is slow but robust. It takes fourteen 8-bit-wide write cycles from the VMEbus computer to completely transfer a single 32-bit word and issue a strobe. Some of the cycles contain data to be loaded in the middle board, some instruct the middle board to transfer its data to one of the end boards, and some contain instructions for one of the end boards.

C.3 Technical Description

The middle board has one 10-bit register, while each end board has one common 32-bit register and two output 32-bit registers. Loading a 32-bit word into an output register of an end board takes 14 write cycles (see table C.1). First, the 32 bits must be loaded onto the common register of the destination end board. This takes a total of 12 cycles, or 3 cycles per byte: 2 to load a byte into the middle board and 1 to transfer it from the middle board to the correct location in the end board. Once the common register of the end board is fully loaded, the 32 bits are latched into one of two output registers, and then a strobe is generated on the end board to signal the remote equipment that the 32-bit output register has changed and must be read anew. Tables C.2 and C.3 show the exact protocol.

A number of write cycles can be eliminated when less than 32 bits of data are needed by one end board. For example, if only 8 bits are required, the total number of write cycles necessary will be five: three to load the 8 bits of data into the common register of the end

Table C.1: An Example

An example of the transfer of the 32-bit word <11000011 01010101 00110011 00001111> to end board *xxx*, output register *y*. The actual meaning of the 32-bit word is irrelevant here, since it is up to the instrument connected to the end board to interpret it. Some of the bits have been grouped to highlight their meaning. Keep in mind that the software sending the 14 bytes must completely invert all 14×8 bits, that is, replace all the 1's with 0's and vice versa, because of an error in the layout of the PIO board.

PIO—Middle								Middle—End Board *y*										Meaning
7	6	5	4	3	2	1	0	2	1	0	9	8	7	6	5	4	3	2	1	0	
0	0	0	0	1	1	1	1														Load bits 4–0
0	0	1	0	0	0	0	0														Load bits 7–5
1	0	0	0	0	*x*	*x*	*x*	0	0	0	0	0	0	0	0	0	1	1	1	1	Transfer bits 7–0 to *xxx*
0	0	0	1	0	0	1	1														Load bits 12–8
0	0	1	0	1	0	0	1														Load bits 15–13
1	0	0	0	0	*x*	*x*	*x*	0	0	0	0	1	0	0	1	1	0	0	1	1	Transfer bits 15–8 to *xxx*
0	0	0	1	0	1	0	1														Load bits 20–16
0	0	1	1	0	0	1	0														Load bits 23–21
1	0	0	0	0	*x*	*x*	*x*	0	0	0	1	0	0	1	0	1	0	1	0	1	Transfer bits 23–16 to *xxx*
0	0	0	0	0	0	1	1														Load bits 28–24
0	0	1	1	1	1	1	0														Load bits 31–29
1	0	0	0	0	*x*	*x*	*x*	0	0	0	1	1	1	1	0	0	0	0	1	1	Transfer bits 31–24 to *xxx*
1	0	0	1	*y*	*x*	*x*	*x*	0	1	*y*											Transfer *xxx* bits 31–0 to *xxx*:*y*
1	0	1	1	*y*	*x*	*x*	*x*	1	1	*y*											Cycle strobe *xxx*:*y*

board, one to copy the common register to one of the output registers, and a final one to issue a strobe.

Along the same lines, some of the 14 write cycles can occur out-of-order. For example, one could load the high 5 bits of the middle board register before the 5 low bits, or one could transfer the four bytes to the end board in a different order, as long as they are properly indexed. In general, however, it is simpler to have a simple software routine that always transfers 32 bits and in a fixed order. Then in order to transfer any number of bits, one can just call that routine with a 32-bit number (most of which can be zeroes), an end board number and an output register number.

The 32 bits of each output register are active-high, while the strobe is active-low. The register data is guaranteed to remain stable throughout the high-low-high strobe cycle, and therefore either the high-low or low-high transitions of the strobe can be used in the remote module as signals that the register must be re-read.

The connection from the PIO board to the middle board is a 50-wire ribbon cable. In contrast, the cables connecting the middle board and the end boards are shielded and twisted 16-wire, to avoid interference. This was considered necessary because twisted cables are more manageable and because some of the cables might need to be up to ten feet long.

Table C.2: Transfers Between the PIO Board and the Middle Board
This table shows in detail the meaning of the 8-bit word sent by the PIO board to the middle board of the digital interface. The 10-bit register of the middle board is loaded in two write cycles, each of which loads 5 bits. When the 10-bit register in the middle board is loaded, its contents are sent to end board x along with 3 control bits (that are actually bits DATA5-DATA3). The 10-bit data contains 2 bits of address plus 8 bits of data, the 2 address bits telling which one of four data bytes is being transferred. Transferring 8 bits of data to the common register of the end board takes 3 write cycles, and this must be repeated four times for a 32-bit word. Once the common register of end board x is fully loaded, its 32-bits are transferred onto output register y of that end board, and then strobe y is activated.

PIO data bits								Meaning to the middle board
7	6	5	4	3	2	1	0	
0	0	0	←——5 bits ——					Load low 5 bits into 10-bit register
0	0	1	←—2+3 bits —					Load high 5 bits into 10-bit register
1	0	0	0	0	—x —			Send 10-bit register to x
1	0	0	1	y	—x —			Tell x to transfer 32-bit data into output register y
1	0	1	1	y	—x —			Tell x to cycle strobe y

Table C.3: Transfers Between the Middle Board and the End Board
This table shows in detail the meaning of the 13-bit words sent by the middle board to one of the end boards (end board x, output register y in this example). The 3 highest bits are control bits, the next 2 are address bits and indicate which data byte is being transferred, and the last 8 bits are data. The entire transfer takes 6 cycles (actually 14 cycles from the point of view of the computer): first the four 8-bit words must be loaded into the common register of the end board, then they are latched into one of the two output 32-bit registers, and finally a strobe is issued to signal the remote instrument that the 32-bit word is available.

CTRL			10-bit DATA										Meaning to the end board
2	1	0	9	8	7	6	5	4	3	2	1	0	
0	0	0	0	0	————bits 7–0 ————								Load data bits 7–0
0	0	0	0	1	————bits 15–8 ————								Load data bits 15–8
0	0	0	1	0	————bits 23–16 ————								Load data bits 23–16
0	0	0	1	1	————bits 31–24 ————								Load data bits 31–24
0	1	y	————not decoded ————										Transfer 32-bit word into register y
1	1	y	————not decoded ————										Cycle strobe y

C.4 History of the Digital Interface

The version of the digital interface that is described here was designed and built in record time for the first observing run at Parkes, in Australia. A sacrifice we made for the sake of design speed was to be able to send information out to the remote modules, but not being able to read them. In retrospect, having the capability to read them would have been very useful, for now we can write to them but we can't even tell from the computer whether they are connected or not. The next generation of the digital interface would have to include two-way communication for feedback.

Since only one middle board is necessary, it is wire-wrapped. Despite the fact that the data protocol of the digital interface supports up to 8 end boards, the current middle board only has 5 output connectors because that is all the FPTM needs. For the dual frequency upgrade of the FPTM, a new middle board with 8 outputs will need to be constructed and will replace the current 5-output box.

The end boards are printed into small PCBs that reside inside the remote box, and take 5 Volt power from the remote box itself. Installing and end board in a new instrument is a simple matter, and most of the effort goes into deciding which bits will control what and writing the control software.

Note that the PIO board logically inverts the 8 data bits that go through it (see appendix B), and that therefore the software must invert them before sending them to the PIO board to make sure that they are interpreted correctly by the middle and end boards.

Caltech graduate student Jagmit Sandhu wire-wrapped the current middle board and also laid out the end boards for the fabrication of the PCBs.

Appendix D

The Buffer Board

Abstract

The Buffer Board is a 6U VMEbus board that can be used to read and buffer digital data into a VME computer, or onto a hard disk or tape connected to the computer. The input data to the board consists of 8, 16 or 32 bits and a strobe, all TTL levels. The input data is automatically written into one of two large memories. When that memory becomes full, a VMEbus interrupt is issued to the computer and the new incoming data is routed to the other memory. The memory not being written to can be read by the VME computer, asynchronously with the incoming data.

D.1 The Front-Panel Input

The input data can be 8, 16 or 32 bits wide, and requires a strobe or clock that indicates when the data bits must be sampled. All data bits as well as the strobe are TTL signals, and bits which corresponds to a HIGH TTL level will be read as 1's, and 0's otherwise.

The relative timing of the strobe with respect to the data is not important as long as it remains stable, since the strobe can be delayed on-board with respect to the data until the timing is correct. The exact delay that can be applied to the strobe is software selectable from 0 to 480 ns in units of 60 ns. Likewise, whether the rising or falling edge of the strobe must be used can be selected from the software.

The maximum sustained strobe rate is better than 1.5 μs, but it really depends more on the speed at which data can be read from the buffer board and written to permanent storage. In burst mode, the separation between strobes can be as small as 200 ns, as long as the average strobe rate is larger than 1.5 μs. This is a hard limit and cannot be reduced to less than 200 ns without changing components in the buffer board itself.

In the current printed circuit-board version of the Buffer Board, there is a front panel 40-pin ribbon cable connector. This connector contains 16 pins for input and 1 for the strobe, as well as multiple grounds. This means the board can be used as is for 8 and 16-bit parallel input, but requires special provisions for 32-bit wide input.

D.2 The VMEbus Output

Whether the input is 8, 16 or 32 bits wide, the Buffer Board is always read through the VMEbus in 32-bit words. This allows for a minimum number of transfers and least overhead on the computer end. In the case of 8- and 16-bit input, earlier data are written in the least significant part of the 32-bit word. Thus data in higher memory positions always correspond to later times.

The rate at which data can be read out of the board and into the VME computer depends mostly on the speed of the device the data are written to, say a hard disk or tape drive. For a 1 GB type hard disk, we have achieved sustained data rates of 1.5 MB/s, although of course most transfers happen in bursts when the CPU takes care of its most pressing needs.

D.3 Memory Switching

The size of each memory bank is always 256 kB, independent of the input bus width. As a result, an interrupt will *always* mean that 64k words are waiting to be read. The computer cannot choose which memory bank to read; it will automatically read the one that is not being written to. This means that if it cannot read the entire bank before the next interrupt arrives, the data not read will be lost. Therefore, it is important to respond to interrupts quickly, as is expected in a real-time system, or else data will be lost.

D.4 Selftest Features

The Buffer Board has some built-in self-test features. The most important one is that the board is able to generate test data. The test data is generated as digital output on a ribbon cable, and a 40-pin ribbon cable loop can feed the test data into the board input. In this manner the entire input circuit of the board can be tested, and one can see whether the outpout data is read correctly and the interrupts received when they are expected.

The test data can be generated a word at a time, or a data word can be set and then a strobe rate chosen. The former method is preferred when data integrity is to be tested, the latter when interrupts are the issue.

For this test mode the input part of the board is completely unaware that test data is being read, since it comes from the outside just like read data. Another test mode is available, where the board can force some of the input bits to 0's or 1's and thus test the memories without having to produce output test data. A loop-back ribbon cable (or a ribbon cable with real data) is still needed in order to produce an input strobe.

Needless to say, the digital test data produced by the Buffer Board can be used as digital data for any other purpose. With the current PCB, 16 bits and a strobe are generated.

Table D.1: The Control Register

The 16-bit control register defines the behavior of the buffer board. Bits 0–7 specifies how the input data and strobe are to be interpreted. Bits 8-15 are used to generate test data, which can be fed back into the buffer board with a ribbon cable loop.

Bits	Function	Values
1,0	input bus width	00= 8-bit, 01= 16-bit, 11= 32-bit
2	strobe polarity	0= rising edge, 1= falling edge
3	overwrite test mode	enables test data. 0= off, 1= on
7-4	on-board strobe delay	0000= none, 0001= 60ns, 0010= 120ns ...1000= 480ns
11-8	output strobe period	0000= none, 0001= 1μs, 0010= 2μs ...0101= 16μs
12	overwrite test data	value of data bits 0,1,8,9,16,17,24,25
13	overwrite test data	value of data bits 2,3,10,11,18,19,26,27
14	overwrite test data	value of data bits 4,5,12,13,20,21,28,29
15	overwrite test data	value of data bits 6,7,14,15,22,23,30,31

The timing of this strobe with respect to the data cannot be fiddled with, since that option exists at the input end alone.

Note: In self-test loop-back mode the board will try to generate an interrupt immediately after it sends a test word out, and sometimes this results in a missing interrupt. It is a problem with the interrupt timing and it can be fixed by modifying the VMEbus interface PALs.

D.5 Internal Registers

The Buffer Board contains a single **16-bit control register**. This control register controls the input bus width, the strobe polarity and delay, and the test data. The meaning of the 16 bits of the control register is explained in table D.1.

Note: It is important that in normal operation bit 3 (the overwrite test mode bit) be zero, or else some of the input data bits will be overwritten to 0's or 1's (depending on control register bits 15-12).

In addition to the control register, the Buffer Board has a **16-bit data register**. In order to generate fake data, one must first write the data into the data register, then generate an output strobe. That strobe will clock the data into the input part of the board if a loop-back ribbon cable is connected.

A third register is available, which contains the **16-bit address** where the next 32-bit input word will be written. Thus, this 16-bit value can be used as a counter that tells how much data has been written into the current input memory bank, and how much time there is before the next interrupt is issued. Note that the address counter could be in a transitional state when it is read from the CPU (since the two processes are asynchronous),

Table D.2: Internal Functions of the Buffer Board

#	Data	Function	Description
0	none	board reset	resets internal counters to bank A and address 0
1	16-bit	write control register	writes a 16-bit word to the control register
2	16-bit	write data register	writes a 16-bit word to the data register
3	none	test clock	generates output strobe pulse
4	16-bit	read data	reads a 32-bit data word from memory
5	16-bit	read counter	reads the 16-bit address of the current input word

Table D.3: Buffer Board Addressing

Bits	Description
31-24	select the board. For example, 0xEB
23	confirms that a VMEbus READ is being done. 0= write, 1= read
22-21	selects one of four function chips. Always 00 on the Buffer Board
20-18	selects a function. 0=reset, 1=write control register, etc (see above)
17-02	16-bit address point to the 32-bit word to be read
01	ignored by the Buffer Board
00	is not part of the VMEbus

and therefore you should either take its value with a grain of salt or read it several times for confirmation.

D.6 Internal Functions

In order to read a word from the output memory bank into the computer, or set the control register, or write to the output register, one must access one of the board functions from the VMEbus. The six available functions are described in table D.2.

D.7 CPU Control of the Buffer Board

Despite the fact that some of the data transfers over the VMEbus are only 16 bits wide, like the address counter reads, all data transfers over the VMEbus should be 32-bit wide. The current Buffer Board will respond to a bus read or write when the 8 most significant bits match those in a board PAL (for example, A31-24=0xEB). The rest of the address bits select the internal board function, while the data to be written or read will be transferred over the 32-bit VME data bus.

The meaning of the 32 bits of address space in the buffer board is described in table D.3. The reason for bit 23 is now obsolete, but for historical reasons it must still be set to 1

when the Buffer Board is to write a word on the VME data bus. One can consider this bit as part of a 4-bit board function address (instead of just 3 bits), which does not result in any overhead.

The following piece of code resets the board, sets the control register to 16-bit input bus, then reads the internal address counter.

```
unsigned long u;
*(unsigned long*) 0xEB000000 = 0;        /* reset board */
*(unsigned long*) 0xEB040000 = 1;        /* set CTRL to 16-bit */
u = *(unsigned long*) 0xEB940000;        /* read address counter */
```

D.8 History

The Buffer Board was originally designed for pulsar radio astronomy involving Mark III VLBI recorders. The observer was to record the data on a Mark III tape, then bring it back to Caltech and read it using an especially modified Mark III vlbi recorder. The recorder was modified by Dave Fort so that it would produce a digital output stream corresponding to the digital data in one of the Mark III channels.

A one-channel data stream consists of 8 bits, in TTL levels and differential mode, as well as a strobe. Due to the relatively high data rate (0.5 MB/s), some method of buffering was required. Hence the first Buffer Board, which was wirewrapped, had 8-bit differential input.

Next came the printed circuit board version (PCB Rev 0) of the Buffer Board, with few changes. It was soon realized that the data read through the Buffer Board contained too many high bits because the memories were driving the VMEbus directly, without the help of drivers.

Therefore a second printed circuit board was designed (PCB Rev A), this time with single-ended 16-bit input. Drivers were added, the address counter was made available to the computer and the internal logic was improved. The differential input connectors were removed because this board was now intended for a different project, the Flexible Filterbank, where a 16-bit data bus is used instead of differential 8-bit.

Note that the board can still be used in 8, 16 and 32-bit mode, as well as differential, but in some cases an extra connector will have to be added, or a small daughter board with differential receivers.

The next PCB version should be able to produce and receive 32-bit data by using both connectors for both input or output, should have a presettable counter to limit the size of the memories to less than 64 kW, and should have user-readable data and control registers. Perhaps it should also provide 16-bit memory output for 16-bit VMEbus.

D.9 Modifications to the Rev.A PCB

The Revision A printed circuit board has to be modified for proper operation. The modifications are the following:

1. Solder a 14-pin socket into TB1 and a jumper between pins 2 and 13. This will insure that the board responds to a VMEbus cycle.
2. Solder a 24-pin socket into one of the U*** spots for the VPULSE PAL.
3. Insert the following PALs: VVME1 into U57, VVME2 into U56, VVME3 into U58, VMUX into U14 and U16, VCOUNT into U18 and VPULSE into U***.
4. Make sure that the two 74AS867 in the address counters are of the AS family and not the ALS family. The ALS chips sometimes generate double interrupts for unknown reasons.
5. Add a wire between U26(1) and U27(15). This provides the `RESET*` signal to the AS867 counters.
6. Add a wire between U30(2) and U19(10). This provides `DW0*` to the VDATA PALs.
7. Add a wire between U30(5) and U19(11). This provides `DW1*` to the VDATA PALs.
8. Provide power to the VPULSE PAL: connect U***(12) to U1(12) and U***(24) to U1(24).
9. Provide the three inputs to the VPULSE PAL: `VMECLK` from U15(8) to U***(1), `DCLK` from U18(23) to U***(2), `RESET*` from U17(1) to U***(13).
10. Get the output from the VPULSE PAL: `PULSE*` from U***(14) to U18(11).

Appendix E

The FPTM at Parkes

E.1 Introduction

This appendix collects a few passages of interest that do not merit the dedication of a chapter or an appendix. For this reason it is somewhat disjoint, and will be mainly of use as a reference.

E.2 AC Power

The FPTM requires AC power of 110 V, exclusively, and consumes a total of 3 kW. The power budget is shown in table E.1. Special care must be kept to ensure that all the racks are properly grounded to each other and to the uninterruptible power supply (UPS), to avoid potentially damaging ground loops. At Arecibo we had the ground straps throwing sparks, and up to 0.2 V of potential difference between the machine ground and the shield of the BNC cables carrying the IF signals.

Wall sockets in Australia carry 240 V and have grounds that are different from the UPS ground. Therefore, one must be careful when using Australian test equipment to look at signals in the FPTM. Our experience was that most sockets were OK to use most of the time, but that sometimes the grounds would be different by as much as 120 V.

E.3 Signal Termination

The telescope signals must also be terminated properly when connected to the FPTM. The 1 second and 5 MHz references have not given any trouble at Parkes because the ground problems are minimal. The two IF signals are run through DC blocks at the input of the upconverter to isolate the mixers in the upconverter, which cannot take *any* DC component in the IF signals.

Table E.1: AC Power Budget

The power requirements of the major subsystems in the FPTM. The total power budget exceeds 3 kW, and is obtained through three separate power points, each from a 1 kW 200—110 V transformer. Modules are grouped for optimal power distribution and only one of the transformers (#3) is taxed beyond its nominal capacity.

Module	Requirements	Transformer
IF Subsystem	3 A	#1
Digitizer	2 A	#2
Correlator	12 A	#3
Pulsar Backend	6 A	#1
Fan Trays	7 A	#2

E.4 Power Levels

A **decibel** (dB) is a dimensionless unit used to express the ratio of two powers (Young 1979). It is 10 times the base-10 logarithm of the power ratio. Thus, if two power values P_1 and P_2 differ by n decibels, then

$$n = \log \frac{P_2}{P_1}.$$

The **dBm** is an absolute unit of power, measured in terms of decibels with respect to 1 mW. Thus the power dissipated by a signal of 1 V_{RMS} into a 50 Ω resistor is of +13 dBm:

$$10 \log \frac{P}{1 \text{ mW}} = 10 \log \frac{(1\text{V})^2/50\Omega}{1 \text{ mW}} = 10 \log 20 = 13 \text{ dBm}.$$

E.5 Voltage Levels

Some of the chapters in this thesis refer to TTL and ECL voltage levels. Both are specifications to be used in digital circuits, not just for supply of power to digital chips but also for clocks and data signals. Each standard, simplified, specifies a voltage to be used for digital HIGH and a voltage for digital LOW. In reality these voltages are actually the voltage ranges that are acceptable for HIGH and LOW, and they are separated by a threshold or a range of voltages inside of which the digital level is undefined or arbitrary. In addition, each standard specifies how to properly terminate signals and what the maximum loads are, in terms of current, impedance or capacitance.

For bipolar TTL (transistor-transistor logic), a logic LOW corresponds to 0.0 V and logic HIGH is anything above 2.5 Volt. ECL (emitter coupled logic) LOW is anything below -1.5 V, while HIGH is anything above -1.0 V. The exact thresholds and ranges for TTL, ECL and other logic families are given in Horowitz and Hill (1989Horowitz and Hill

1989), page 475.

E.6 The Artificial Pulsar Generator

There are two improvements that could be made to the artificial pulsar. First, it could be made to generate output at frequencies of several hundred MHz instead of at baseband. With this change the artificial pulsar could be inserted into the FPTM at the beginning of the IF chain, and be used to test the entire machine instead of only the digital part. The only objection to this suggestion is cost, especially given the fact that the current artificial pulsar at passband is sufficient for most tests.

The second improvement would be to generate a *dispersed* pulsed signal instead of the undispersed pulsar we now have. The advantage of a dispersed signal is that it places more emphasis on the high lags than an undispersed signal does. However, it is also very hard to generate such dispersed signals.

Appendix F

Glossary

Table F.1

Abbreviation	Description
AC	Alternating Current
ALC	Automatic Level Control
ALU	Arithmetic Logic Unit
AO	Arecibo Observatory
AOC	Arecibo Correlator
ATNF	Australia Telescope National Facility
BB	Baseband
BBC	Baseband Converter
BW	Bandwidth
CDRP	Coherent Dispersion Removal Processor
CIT	California Institute of Technology
CSF	Correlator Sampling Function
CSIRO	Commonwealth Scientific and Industrial Research Organisation
DC	Direct Current
DD	Damour and Deruelle
DDGR	Damour and Deruelle with General Relativity
DCLK	Digitizer Sample Clock
DDS	Direct Digital Synthesis
DM	Dispersion Measure
ECL	Emitter Coupled Logic
EM	Emission Measure
FB	Filterbank
FIFO	First In First Out
FPSM	Fast Pulsar Search Machine
FPTM	Fast Pulsar Timing Machine

continued on next page

continued from previous page

Abbreviation	Description
GB	Green Bank
GBT	Green Bank Telescope
GND	Ground
GPS	Global Positioning Satellite
HP	Hewlett–Packard
HTRP	High Time Resolution Processor
IF	Intermediate Frequency
ISM	Interstellar Medium
ISS	Interstellar Scattering
JB	Jodrell Bank
LO	Local Oscillator
LCP	Left Circular Polarization
LSB	Lower Side Band
MCLK	Correlator Multiply Clock
NAIC	National Astronomy and Ionosphere Center
NCO	Numerically Controlled Oscillator
NFRA	Netherlands Foundation for Research in Astronomy
NRAL	Nuffield Radio Astronomy Laboratories
NRAO	National Radio Astronomy Observatory
NSLO	Nançay Swept Local Oscillator
PAL	Programmable Array Logic
PCB	Printed Circuit Board
PLD	Programmable Logic Device
PPS	Pulse Per Second
PROM	Programmable Read-Only Memory
RAM	Random Access Memory
RAW	Read/Add/Write
RCP	Right Circular Polarization
RF	Radio Frequency
RM	Rotation Measure
RMS	Root Mean Square
ROM	Read-Only Memory
SSB	Solar System Barycenter
SP	Spectral Processor
TOA	Time of Arrival
TTL	Transistor-Transistor Logic
SCLK	Correlator Shift Clock
SNR	Signal to Noise Ratio
SSB	Single Side Band
TDB	Barycentric Dynamical Time

continued on next page

continued from previous page

Abbreviation	Description
TDT	Terrestrial Dynamic Time
TOA	Pulse Time of Arrival
UPS	Uninterruptible Power Supply
USB	Upper Side Band
UT	Universal Time
UTC	Universal Coordinated Time
VLA	Very Large Array
VLBA	Very Long Baseline Array
VLBI	Very Long Baseline Interferometry
WSRT	Westerbork Synthesis Radio Telescope
WENSS	Westerbork Northern Sky Survey
WW	Wire Wrap

Bibliography

Backer, D. C., Clifton, T. R., Kulkarni, S. R., and Werthimer, D. J. 1990, *Astr. Astrophys.*, **232**, 292–300.

Backer, D. C. and Hellings, R. W. 1986, *Ann. Rev. Astr. Ap.*, **24**, 537–575.

Backer, D. C., Kulkarni, S. R., Heiles, C., Davis, M. M., and Goss, W. M. 1982, *Nature*, **300**, 615–618.

Backer, D. C., Werthimer, D., Kapadia, A., and Rabaey, J. 1994. in prep.

Bailes, M. *et al.* 1994, *Astrophys. J.*, **425**, L41–L44.

Bailes, M., Manchester, R. N., Kesteven, M. J., Norris, R. P., and Reynolds, J. E. 1990, *Nature*, **343**, 240–241.

Bhattacharya, D. and van den Heuvel, E. P. J. 1991, *Phys. Rep.*, **203**, 1–124.

Blackman, R. B. and Tuckey, J. W. 1958, *The Measurement of Power Spectra* (New York: Dover Publications, Inc).

Boriakoff, V., Buccheri, R., and Fauci, F. 1983, *Nature*, **304**, 417–419.

Camilo, F. 1994, *In preparation.*

Camilo, F., Nice, D. J., and Taylor, J. H. 1993, *Astrophys. J.*, **412**, L37–L40.

Camilo, F., Thorsett, S. E., and Kulkarni, S. R. 1993, *Astrophys. J.* in press.

Clark, T. A. and Rogers, A. E. E. *Mark III VLBI Data Acquisition Terminal.* Goddard Space Flight Center, Greenbelt, Maryland 20771, 1982.

Cole, T. 1968, *Aust. J. Phys.*, **21**, 273–282.

Cooper, B. F. C. 1970, *Aust. J. Phys.*, **23**, 521–527.

Deich, W. T. S. 1994. Ph.D. thesis, California Institute of Technology.

Dwarakanath, K. S. and Shankar, N. U. 1990, *J. Astrophys. Astr.*, 323–410.

Ewing, M. S. 1989a. High-Speed Correlator Chip, Final Specification. Technical report, California Institute of Technology, Caltech 105-24, Pasadena, CA 91125.

Ewing, M. S. 1989b. OVRO High-Speed Correlator Crate and Board Hardware Introduction and Overview. Technical report, California Institute of Technology, Caltech 105-24, Pasadena, CA 91125.

Filippenko, A. V., Readhead, A. C. S., and Ewing, M. S. 1983, AIP Conference Proceedings Vol. 101, 113–117.

Fisher, R. *NRAO Spectral Processor Memo 34.* Edgemont Road, Charlottesville, VA 22903-2475 1989. Internal Memo.

Fisher, R. *The Green Bank Spectral Processor.* Edgemont Road, Charlottesville, VA 22903-2475 1994. Internal Memo.

Foster, R. S. 1990. Ph.D. thesis, University of California, Berkeley.

Foster, R. S. and Cordes, J. M. 1990, *Astrophys. J.*, **364**, 123–135.

Foster, R. S., Wolszczan, A., and Camilo, F. 1993, *Astrophys. J.*, **410**, L91–L94.

Fruchter, A. S., Stinebring, D. R., and Taylor, J. H. 1988, *Nature*, **333**, 237–239.

Hagen, J. *Instruction Manual. Project Number 579. 40 MHz Correlator.* P.O. Box 995, Arecibo, PR 00613 1987. Manual No. 8319, Rev. 1.

Hales, S. E. G., Baldwin, J. E., and Warner, P. J. 1993, *Mon. Not. R. astr. Soc.*, **263**, 25–30.

Hartmann, D., Epstein, R. I., and Woosley, S. E. 1990, *Astrophys. J.*, **348**, 625–633.

Hewish, A., Bell, S. J., Pilkington, J. D. H., Scott, P. F., and Collins, R. A. 1968, *Nature*, **217**, 709–713.

Horowitz, P. and Hill, W. 1989, *The Art of Electronics* (Cambridge: Cambridge University Press), 2nd edition.

Johnston, S. and Bailes, M. 1991, *Mon. Not. R. astr. Soc.*, **252**, 277–281.

Johnston, S. *et al.* 1993, *Nature*, **361**, 613–615.

Kaspi, V. M., Taylor, J. H., and Ryba, M. 1994, *Astrophys. J.* submitted.

Kulkarni, S. R. 1992, *Phil. Trans. R. Soc. Long. A*, **341**, 77–92.

Lestrade, J. F., Bougois, G., Biraud, F., Aubry, D., Drouhin, J. P., and Freon, G. 1990, in *Impact of Pulsar Timing on Relativity and Cosmology*, ed. D. Backer (Berkeley: Center for Particle Astrophysics), E1–E10.

Lorimer, D. R. 1994, *in preparation.*

Lorimer, D. R. *et al.* 1994, *Astrophys. J.* Submitted.

LSI Logic Corp., Milpitas, CA 95035. *L1A1766* 1999.

Manchester, R. N. and Taylor, J. H. 1977, *Pulsars* (San Francisco: Freeman).

McKinnon, M. M. 1992. Ph.D. thesis, New Mexico Institute of Mining and Technology.

Motorola Inc., Santa Clara, CA. *VMEbus Specification Manual* 1985.

Motorola Inc., Santa Clara, CA. *MCA2500ECL Macrocell Array Design Manual* 1986.

Nice, D. J., Taylor, J. H., and Fruchter, A. S. 1993, *Astrophys. J.*, **402**, L49–L52.

Padin, S. *et al.* 1993, *IEEE Trans.*, **42**, 793–798.

Padin, S. and Ewing, M. 1989, *IEEE Trans. Instrum. Meas.*, **38**, 1109–1115.

Padin, S., Scott, S., Woody, D., Scoville, N., Seling, T., Finch, R., Giovanine, C., and Lawrence, R. 1991, *Publ. Astr. Soc. Pacific*, **103**, 461–467.

Phillips, J. A. 1991. Ph.D. thesis, Cornell University.

Phinney, E. S. 1992, *Phil. Trans. R. Soc. Lond. A*, **341**, 39–75.

Phinney, E. S. and Kulkarni, S. R. 1994, preprint.

Rankin, J. M., Stinebring, D. R., and Weisberg, J. M. 1989, *Astrophys. J.*, **346**, 869–897.

Rupen, M. P., Vangorkom, J. H., Knapp, G. R., Gunn, J. E., and Schneider, D. P. 1987, *Astron. J.*, **94**, 61.

Ryba, M. F. 1991. Ph.D. thesis, Princeton University.

Salter, M. J., Lyne, A. G., and Anderson, B. 1979, *Nature*, **280**, 477–478.

Segelstein, D. J., Rawley, L. A., Stinebring, D. R., Fruchter, A. S., and Taylor, J. H. 1986, *Nature*, **322**, 714–717.

Shapiro, I. I. 1964, *Phys. Rev. Lett.*, **13**, 789.

Shklovskii, I. S. 1970, *Sov. Astron.*, **13**, 562–565.

Simard-Normandin, M., Kronberg, P. P., and Button, S. 1981, *Astrophys. J. Supp. Series*, **45**, 97–111.

Stinebring, D. R., Kaspi, V. M., Nice, D. J., Ryba, M. F., Taylor, J. H., Thorsett, S. E., and Hankins, T. H. 1992, *Rev. Sci. Inst.*, **63**, 3551–3555.

Stinebring, D. R., Ryba, M. F., Taylor, J. H., and Romani, R. W. 1990, *Phys. Rev. Lett.*, **65**, 285–288.

Tauris, T. M. *et al.* 1994, in press.

Taylor, J. H. and Cordes, J. M. 1993, *Astrophys. J.*, **411**, 674–684.

Taylor, J. H., Hulse, R. A., Fowler, L. A., Gullahorn, G. E., and Rankin, J. M. 1976, *Astrophys. J.*, **206**, L53–L58.

Taylor, J. H. and Weisberg, J. M. 1989, *Astrophys. J.*, **345**, 434–450.

Thompson, A. R., Moran, J. M., and Swenson, G. W. 1991, *Interferometry and Synthesis in Radio Astronomy* (Malabar, Florida: Krieger Publishing Company).

Thorsett, S. E. 1991. Ph.D. thesis, Princeton University.

Thorsett, S. E., Arzoumanian, A., McKinnon, M. M., and Taylor, J. H. 1993, *Astrophys. J.*, **405**, L29–L32.

Wolszczan, A. 1990. IAU circular 5073.

Wolszczan, A. 1991, *Nature*, **350**, 688–690.

Wrobel, J. M. 1991. Very-Long-Baseline Interferometry at the Very Large Array: II. The Long Guide. Technical report, National Radio Astronomy Observatory, P.O. Box 0, Socorro, NM 87801.

Young, F. C. 1979, *Dictionary of Electronics* (Middlesex, England: Penguin Books).