CHAPTER TWO

DESIGN PHILOSOPHY

In most cases, the design of a signal processing instrument gets greatly simplified if some properties of the signal are already known, because many redundancies and unwanted operations can be avoided. This results in arriving at a more reliable, faster and cost effective solution, when the architecture can be tuned for a particular set of tasks instead of being general purpose in nature. While this philosophy can be applied to a large extent, but there must be some room left behind in the design to provide the flexibility for future enhancements. Keeping this in view, many characteristics were taken into account such as the intrinsic nature of pulsar signals, models of the propagation effects of the interstellar medium and the earth's ionosphere, nature of local interference, response of antenna arrays, behavior of receiver systems, behavior of digital hardware at high-speed, etc., to arrive at suitable numerical algorithms for computation. Information from all these factors were used in arriving at the optimal design. This chapter presents the various factors that were considered in achieving the simplifications. The optimizations are classified into two groups - those which simplify signal processing algorithms, and those which simplify the hardware/software implementation.

2.1 Signal Processing Considerations for System Design:

In this section, the signal processing requirements and optimizations done for different types of pulsar observations are discussed. The discussion will present the scenario in general, but will emphasize on those optimizations relevant to the GMRT telescope. Classified broadly, observations of pulsars fall into two major categories:

a. Observations conducted to search and discover unknown pulsars.
b. Observations to study the emission properties of the pulsar and the effects of the intervening medium, by analyzing the intricate structure of the pulse with time.

These two types of observations require different approaches in signal processing as elaborated below.

2.1.1 Signal Processing for Pulsar Search:

Detection of weak pulsar signals that are generally buried deep in noise requires special signal processing techniques especially when none of the parameters like flux density, period, extent of dispersion, degree of Faraday rotation, Doppler accelerations local to the pulsar environment are known apriori. One has to search in the multi-dimensional parameter space in fine steps to look for a combination of the parameters that fits the observed data best. The best fit parameters can then be examined to see whether there are any significant deviations beyond the statistical fluctuations due to random noise, that can be qualified as a valid detection. This needs extensive computations (of the order of several Gflops per field in the survey), which is usually iterative and user-interactive. Hence it is preferred to record the data in as raw a form as possible leaving the rest of the computations for off-line processing. However the observing bandwidths are usually large, resulting in large data rates, of the order of several Msamples per second. This makes some on-line pre-processing desirable, and at times essential, to reduce the data rate to proportions palatable to the data recording systems. A careful selection of the parameters that influence the detection sensitivity is however, to be ensured such that the probability of detection is not compromised on, while minimizing the complexity in design.

2.1.1.1 Sensitivity Considerations at Radio Wavelengths:

Usually pulsar surveys are conducted using telescopes with very high sensitivity The reason is obvious, looking at figure (2.1) where the flux densities of pulsars are displayed. The telescope system receives the pulsar signal amid radiation from several other terrestrial and celestial sources falling within the beam of the antenna. Besides, the receiver also contributes some noise in the amplifier and mixer stages. Expressing the noise power in equivalent temperature, the noise contributed by the sky can be designated as T_{sky} and that from the receiver as T The signal from the pulsar causes an enhancement in the sky noise temperature by an amount T_a, called the Antenna temperature. This temperature is proportional to the flux density of source and is related as

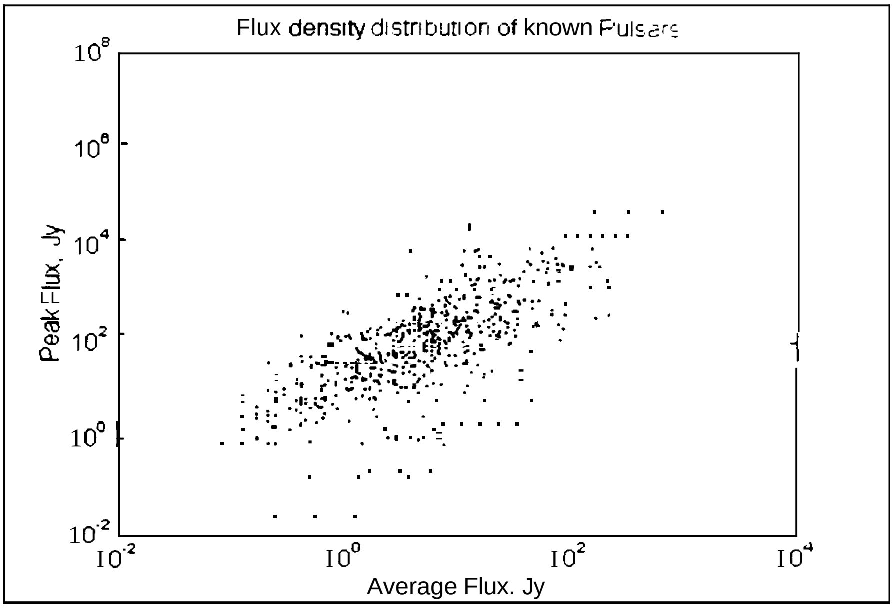

Fig. *2.1*

$$T_a = \frac{A_e S_{pulsar} \cdot 10^{-26}}{2K_b} , {}^\circ K \qquad (2.1)$$

where A_e - effective collecting area of the telescope

K_b = Boltzman constant

S = Source flux in Jansky (10^{-26} $Wm^{-2}Hz^{-1}$)

The signal to noise ratio at the input of receiver is then given as

$$\mathbf{SNR} = \frac{T_a}{T_{rx} + T_{sky}} \tag{2.2}$$

The presence of the power received from the pulsar can then be found by comparing the mean power received during the on-pulse (μon) and off-pulse regions(μoff), in terms of the rms off-pulse noise power(σ off), as

$$\mathbf{SNR} = \frac{\mu_{on} - \mu_{off}}{\sigma_{off}} \tag{2.3}$$

The base-band is usually sampled at Nyquist rate. To reduce the data recording rate, some samples may be pre-integrated over a time interval τ_{int} The time constant of this integration is chosen to be much less than the pulse width. For a given channel bandwidth Δ_f, the samples of noise separated by a time interval $1/\Delta_f$ or more are statistically independent. The number of independent samples, N, over a time constant τ_{int} can then be expressed as $N = \tau_{int}.\Delta_f$.

If N independent time samples are added together, then the signal-to-noise ratio grows by $\sqrt{N}$. If two orthogonal polarization channels are available, their contribution can also be added after total-power detection. If N_p is the number of polarizations in general, then the minimum detectable change in flux is given by (Kraus, 1966)

$$\Delta S_{min} = \frac{2K_b \ (T_{rx} + T_{sky})}{A_e\sqrt{N_p \cdot \Delta_f \cdot \tau_{int}}} \tag{2.4}$$

In case of pulsars, the optimum τ_{int} is chosen to match the pulse- width (W), which is usually a small fraction of the pulse period. Then the data are recorded with a much finer sampling interval compared to τ_{int} and further processed off-line. During the off-line processing a long stretch of the data may be used to detect the presence of a periodic pattern. Generally, the data stretch is Fourier transformed to produce the power spectra in which the presence of any periodic signal may be detected as enhanced components at the pulse frequency and its harmonics. This is equivalent to synchronously folding the noisy pulse profile over its pulse period all through the stretch of the chosen data., if the contribution at the harmonics of the pulsar frequency harmonics are added coherently. If there are N periods of a pulsar in a time stretch (Tobs) of data chosen for detection, then the minimum detectable peak flux is

$$\Delta S_{min} = \frac{2K_b\ (T_{rx} + T_{sky})}{A_e\sqrt{N_{pol}\ .\ N_{periods}\,\tau_{int}\ .\ \Delta_f}} \qquad (2.5)$$

such that

$$P.N_{periods} = T_{obs} \text{ and } W = \tau_{int}$$

However, it is customary to denote the detection limit in terms of the "average flux" of the pulsar as

$$S_{avg} = \left[\frac{W}{P}\right] S_{peak} \qquad (2.6)$$

If β is the minimum SNR (Signal to noise ratio) required to consider a valid detection then

$$\Delta S_{avg\_min} = \frac{2.\beta.K_b\ (T_{rx} + T_{sky})}{A_e\sqrt{N_{pol}\ .T_{obs}\ .\ \Delta_f}} \cdot \sqrt{\frac{W}{P}} \qquad (2.7)$$

when W << P, but in general, this is shown to be (Vivekanand, Narayan and Radhakrishan, 1982) as

$$\Delta S_{avg\_min} = \frac{2.\beta.K_b\ (T_{rx} + T_{sky})}{A_e\sqrt{N_{pol}\ .T_{obs}\ .\ \Delta_f}} \cdot \sqrt{\frac{W}{P-W}} \qquad (2.8)$$

Typically, the search sensitivity levels are of the order of few mJy. With the ORT, the total integration may be of the order of 10 minutes to reach a sensitivity of few mJy with a bandwidth of 8 MHz. With the GMRT in the incoherent array mode with a bandwidth of 32 MHz , about 3 minutes may suffice to reach same sensitivity. To cover the entire range of pulse periods, it will be sufficient to have a pre-recording integration of about 0.25 millisecond to about 4 millisecond in each of the spectral channels (256 channels with ORT and 512 channels with GMRT).

2.1.1.2 Choice of Optimum Bandwidth:

As explained in section (1.2.1.), the bandwidth of observation is to be optimized to find the balance between the dispersive smearing across the band and the channel time constant (Deshpande, 1989). Using equation (1.9) the optimal total bandwidth for these frequencies for GMRT is given by

$$\Delta f_{tot} = 512\Delta f_{ch\_opt} \qquad (2.9)$$

Figure (2.2a) displays the total optimum bandwidth at each of the GMRT frequencies as a function of the Dispersion Measure, and Figure (2.2b) shows the residual smearing that would exist within a channel after incoherent de-dispersion with 512 frequency channels across the total bandwidth. Of the GMRT frequencies, the most favourable choice for pulsar search would be 1.4 GHz for regions close to the galactic plane and 610/327 MHz for other regions (Deshpande, 1995).

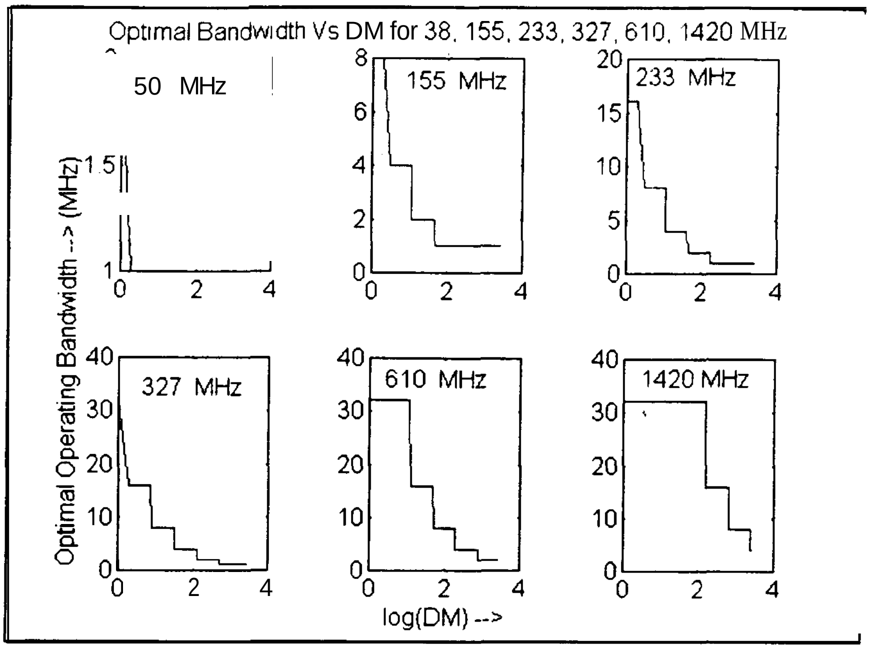

Fig. 2.2a

2.1.1.3 Considerations for On-Line Data Reduction:

The analog base-band signal from the telescope may be chosen to be different depending on the type of observation.

The signal at the input of the AID converter is usually Gaussian random noise, contributed dominantly by the part of the sky visible within the beam of the telescope and the noise generated within the receiver itself. The pulsar signal is usually buried about 20 to 30 dB below the noise level. The AID converter provides 4 bit representation of the samples, and the gain in the front-end of the receiver system is adjusted such that the maximum excursion of the voltages uses the dynamic range of the AID converter optimally (Thompson et., al., 1986). Since the input is dominated by Gaussian noise, it suffices to have the AID input range of about +/- 3σ, (where σ is the rms value of the input) around the mean of the signal(the mean is zero). After detection, the random power signal follows an exponential distribution, wherein the mean is nearly equal to the rms

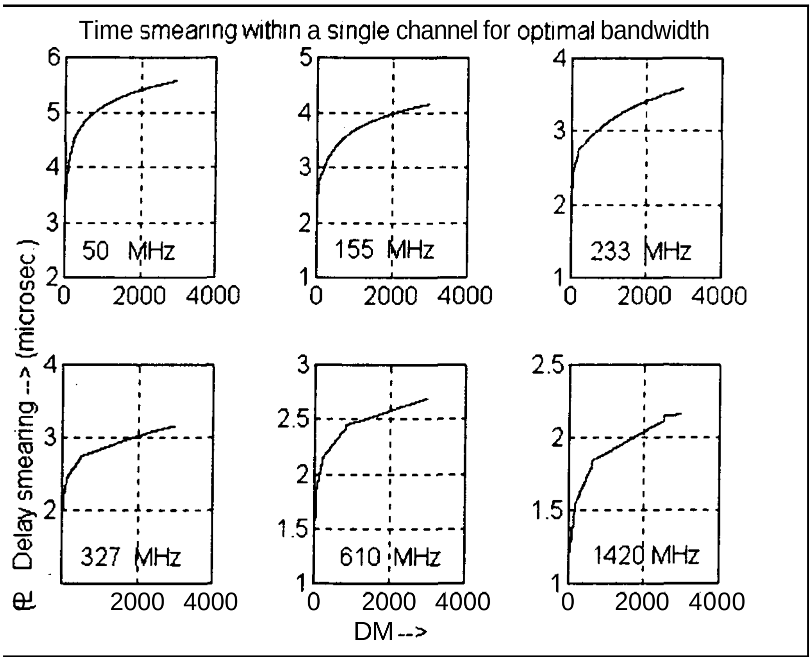

Fig 2.2b

value. For any further integration of Nsamples the mean would grow by N while the rrns grows by $\sqrt{N}$. By dropping $\log_2(\sqrt{N})$ lower bits of the quantized representation, the data width may be reduced without degrading signal-to-noise ratio. Figure (2.3) shows the degradation in SNR as a function of number of bits used to represent the quantized signal (a) power, (b) voltage. This is conveniently used to keep the number of bits at a minimum throughout the rest of the system so as to reduce the number of components and match the standard bit-widths of the available components.

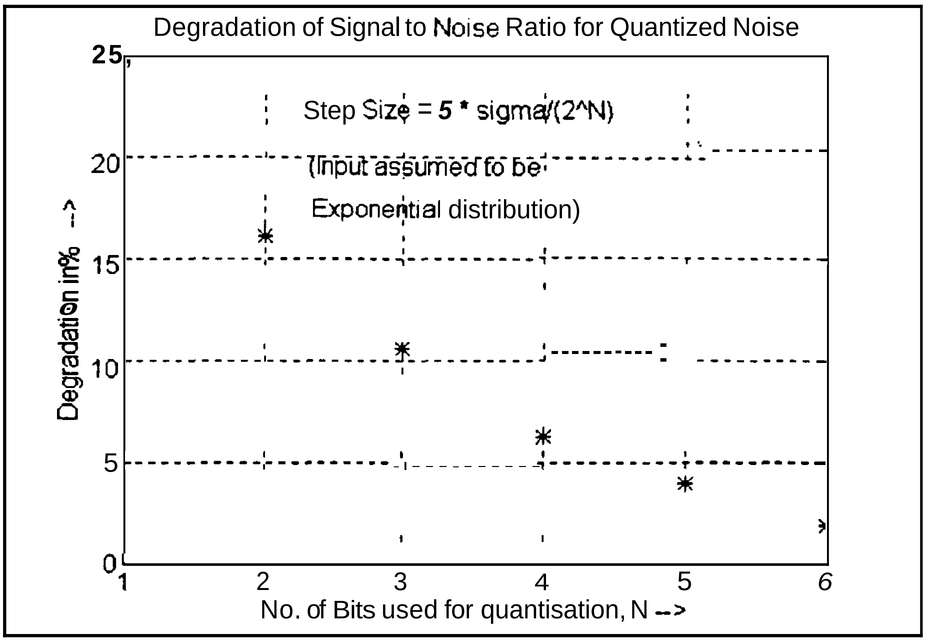

Fig. 2.3 a

At the input of AC, the signal from each dish are represented with 4 bits. When the outputs from a number of dishes say, N, are selected to be summed up, the random noise content in the samples increases by $\sqrt{N}$ in both PA and IA modes(Prabu, 1997), the difference that in the Phased array mode, the sky noise within the main lobe of the phased beam would have added coherently between the dishes and hence would dominate the sky noise component in the summed output, while in the IA mode, summing up the power from all dishes results in the bit growth being dominated by the increase in the 'mean' with the sky noise corresponding to the beam width of a single dish. Thus the maximum growth with all 30 dishes summed up in the PA mode would be $\log_2(\sqrt{N})$ bits, corresponding to about 3 bits, resulting in an overall maximum data width of 7 bits (1 bit sign, 6 bit magnitude) in both real and imaginary terms. For the incoherent array mode, the AC rounds off the

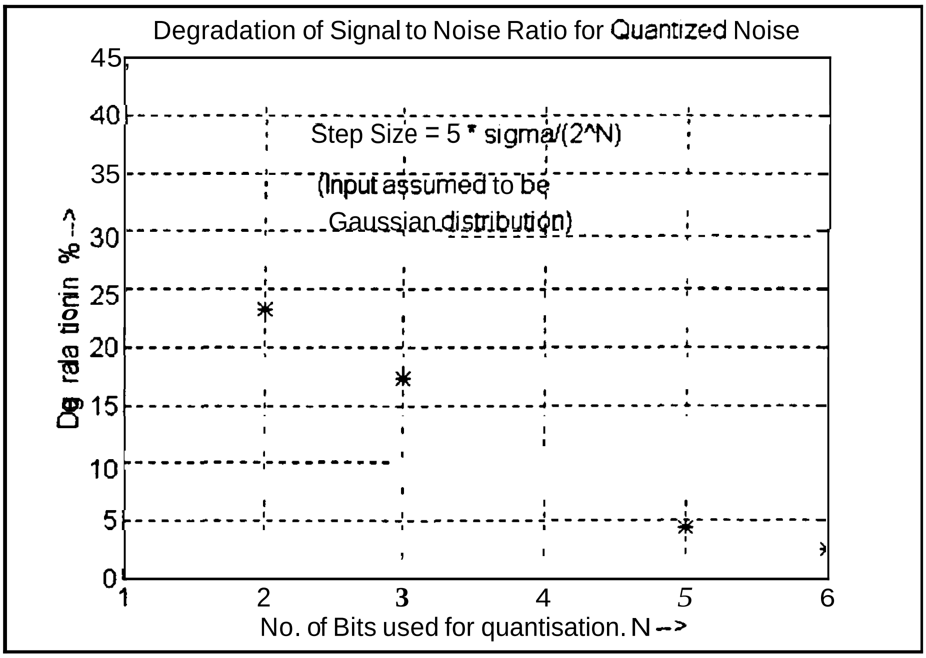

Fig. 2.3 b

detected values to **4** bits before summing the different antenna terms. The bit growth in incoherent mode would be by $\text{Log}_2(N)$ bits corresponding to about 5 bits and an overall data width of 9 bits, (magnitude, no sign bit) when all dishes are selected. These estimates for the PA mode approach those for the IA mode when the signal contribution from the sky within the PA beam becomes dominant. This also gives the observer a choice to sacrifice a few bits and record at a lower comfortable rate. Also, for a given data rate, the saved bit rate can be used to have faster sampling i.e., more bandwidth and hence better sensitivity (Clifton T.R, 1985). But, in most cases, the raw data rate is too high and some samples have to be integrated before recording. As mentioned earlier, the pre-integration time constant may be chosen depending on the on-pulse resolution desired. The pre-integration bit growth of the detected samples is by $\log_2(\text{Nint})$ bits, where Nint is the number of samples integrated. This data may be further quantized to just one or two bits before the final recording. However, some additional care needs to be taken before the final quantization. If only one bit is chosen to represent the sample, it should be used to convey whether the sample value is greater or lesser than the mean value of power. In this case the threshold for I-bit quantization is the mean power itself. If the mean drifts with time due to gain fluctuations of the R_x system, then the threshold has to track the mean suitably. This necessitates a mechanism to evaluate the running mean of the detected power continuously so as to adaptively scale the threshold for quantization.

The running mean estimator and subtractor circuit essentially functions as a high-pass filter, with a cut-off frequency determined by the equivalent time-constant of running integration. This time constant has to be much larger than several pulse periods so that the fluctuations in strength due to the pulse itself doesn't affect the estimation of the running mean. This depends on the pulse energy and the period. In most cases, the average increase in the antenna temperature due to the pulse energy is much less compared to the system temperature and the period ranges between a millisecond to a few seconds. The simplest form of a mean estimator is a FIR filter of N taps with equal weightage for all taps. This forms a rectangular window in time, resulting in a sinc function response in the frequency domain, with the peak of the sinc function falling at zero frequency (the mean) of the signal. As the length of the window is increased, the width of the sinc function narrows, approaching a delta function, and reducing error in the mean estimate as the contribution from other frequency terms are cut down. The choice of a rectangular window results in finite reduction of higher frequency components due to interference from the sidelobes of the sinc function, the maximum reduction being at.a level 13 dB corresponding to the first sidelobe power. The normalized roll-off characteristic is given by

$$H(\omega) = \frac{1}{N}\left[\frac{\sin\left(\frac{NWT_b}{2}\right)}{\sin\left(\frac{WT_b}{2}\right)}\right] \tag{2.10}$$

where T_b is the sampling interval of the N-tap filter and $W=2\pi f$, f is the spectral frequency.

As noted above, the length of the running mean window should be long enough to avoid the pulsar signal from affecting the estimate of the mean. For example, for N=128 and T_b=0.128 s, the leakage of power from the fundamental pulse frequency of a 5 second pulsar period will be about − 23 dB. compared to the mean. In comparison, the statistical uncertainty in computation of the mean is atleast −21 dB at 3σ level. For pulsars with shorter period this contrast is even better. Normally the value of τ_{int} (in on-line processing) is always much smaller than T_b. Thus, an additional integrator is required to integrate (T_b/τ_{int}) pre-integrated sarnples before feeding the FIR filter. The low pass filter output is subtracted from consecutive pre-integrated sarnples, so that the pass band after subtraction is that of a high-pass filter with the cut-off frequency same as the low-pass filter. While tapered windows in the FIR filter could produce sharper rejection, the main lobe width correspondingly increases for a fixed number of taps. Also, in these cases the implementation poses too much of computational load and is usually avoided for on-line processing.

The mean power may fluctuate due to the gain changes of the receiver system and in some cases due to scintillation of a strong source within the field of view while observing pulsars. The estimate of the mean obtained by the low-pass filter must be revised faster than the rate of these variations so that the estimated mean tracks the variations well. While the best choice would be to update the running mean at every raw sampling interval, it is usually sufficient to update the mean at intervals of a small fraction of the length of window without causing significant leakage from aliased components at the output of the low-pass filter.

In multi-bit representation of the final data samples, the final quantization is sensitive to gain differences between the frequency channels due to variations in the receiver band-shape The pass-band ripple of the receiver band-shape is usually less than 3 to 6 dB while the dynamic range of the signal may be much larger after pre-integration. Typically, the maximum available dynamic range of a N bit − quantized data would be about $R=10Log_{10}(2^N)$ dB (the dynamic range is greater than the pass-band ripple for N ≥ 2 for a ripple of 6 dB). These differences have to be equalized before final quantization, so that when the different channel contributions are combined during post processing, they would have same rms fluctuations and can be added with equal weightage. The gain scale factors may be determined from the running means of the channels themselves. After calibration the sign bit can be extracted from each data sample along with a suitable magnitude bit (depending on the signal strength) and recorded. The recorded data is then backed up on a portable medium such as a magnetic tape and ported to the post processing site.

2.1.1.4 Off-line Post-processing:

To reach the required sensitivity levels, the recorded data needs to be processed suitably. If the pulsar period is known, then the data can be integrated synchronously over the pulse period further for very long intervals (several minutes) per beam position. The basic signature that is used to identify a pulsar signal is its periodic train of pulses. When the period is not known, the best technique to look for periodicity in a long stretch of data is Fourier analysis. The data can be Fourier transformed to produce the *fluctuation* spectra , in which the periodic pulse train will show up as enhanced contribution at the pulse frequency and its harmonics. The number of significant harmonics depends on the pulse shape and width. At this stage, the harmonics of a pulsar signal, if it exists, can be added together to improve the signal to noise ratio and hence the detection sensitivity. This process, called harmonic folding, then filters out spurious candidates in the fluctuation spectrum which do not have any harmonic relationships. There are two variations, called incoherent and coherent harmonic folding (Dipankar, 1996), which involve adding together the contribution at the harmonics in the spectrum by combining the powers or by combining the complex voltages with different phase gradients. Coherent folding is more time consuming, while it provides the best sensitivity. As the harmonic folding continues, it is not necessary that the signal to noise ratio (SNR) continues to improve as a function of the number of harmonics used. At some point, the higher harmonics may be so weak that they are insignificant. From then on, further folding will add noise to existing harmonics and only worsen the SNR. In practice, the combination are examined with the number of harmonics increasing in steps of 2 or so. However, the dispersive delay gradient has to be removed before the Fourier transform, since it will otherwise lead to broadening of the pulse and reduce the strength of its spectral features. Since the dispersion measure is unknown to start with, the data of every field has to be de-dispersed for several trial values of DMs, and a set of trial values for which the de-dispersion shows significant improvement in signal to noise ratio in the spectral analysis are selected as probable candidates. While man-made interference fits well only at zero dispersion measure (within a small value of DM), any pulsar signal should be in the domain of higher DM values. So the candidates from local interference can be identified and rejected later. Since this poses a large computing requirement, a short-cut method, called tree-algorithm (Taylor, 1974) is popularly used for de-dispersion. The fluctuation spectra for each of the trial values of DM is examined as described above. The candidates that show up beyond a pre-set detection threshold in the harmonic folding are listed as valid candidates and their DM, period and pulse widths are recorded. In such cases, the complex fluctuation spectrum at the harmonic locations is then Fourier transformed back to time domain giving the average time profile of the pulsar. This processing usually involves a computational rate of the order of several Giga operations per second to be able to achieve the processing in a time comparable to the observation time. Considerable amount of software is already available on standard, high performance computers, which can be used to handle the above tasks for pulsar search post-processing.

2.1.2. Signal Processing for Studies of Pulsars:

Once a pulsar is detected its period, position, dispersion measure, pulse width can all be measured by repeated observations. Then it is possible to estimate the details of the pulse of radiation. Then the spectral and temporal distribution of the pulse energy can be studied to provide interpretation for physical processes occurring at the pulsar and in the interstellar medium. For detailed studies of known pulsars, it is possible to add the signals from all antennas of an array so as to use maximum possible sensitivity. The bandwidth is also maximized depending on the optimal bandwidth considerations explained earlier. Once the parameters such as DM, RM, Doppler acceleration etc., are known it will become possible to correct the distortions of the pulse profile in real time, just as the telescope tracks the pulsar. Some observations may be for rare, one-time investigations. Some observations may be having many unknown parameters such as Doppler accelerations , RM , DM, etc., In both of the above cases, it is preferable to retain the data in as raw as possible, limited only by the recording rate. But in routine observations of a group of pulsars, it will be preferred to reduce the data to a comfortable size and recording rate, thus minimizing the off-line processing. It is preferred in many studies that the data before recording be reduced to a level that it is ready for interpretations during off-line post-processing. Easier said than done, the instrumentation required for such on-line processing is quite complex and needs to be extremely fast and flexible to cater for different types of studies associated with pulsars. The signal processing requirements for different types of observations are discussed below:

2.1.2.1. Full Polarization, Single-Pulse Studies:

Single pulse profile studies are aimed at studying the pulse to pulse variations in the intensity and polarization properties in order to relate these features to the changes in the pulsar radiation and its environment. This is used in study of sub-pulses and micro-pulses, which define the pulse shape. These types of observations demand fast sampling (limited only by dispersive smearing). These studies are possible when the pulsar signal is strong enough to be visible with minimum integration (i.e., at single pulse level). However, one can use large bandwidth for such observations to improve the sensitivity. This will require that the dispersion and rotation measures are known so that the dispersive delay gradient in arrival time of the pulses at different frequencies and the distortion of polarization parameters due to Faraday rotation can be calculated and corrected. Subsequently all the frequency channels of the band can be summed on to a single time sequence in order to improve the signal to noise ratio and to reduce the associated data rate and memory size.

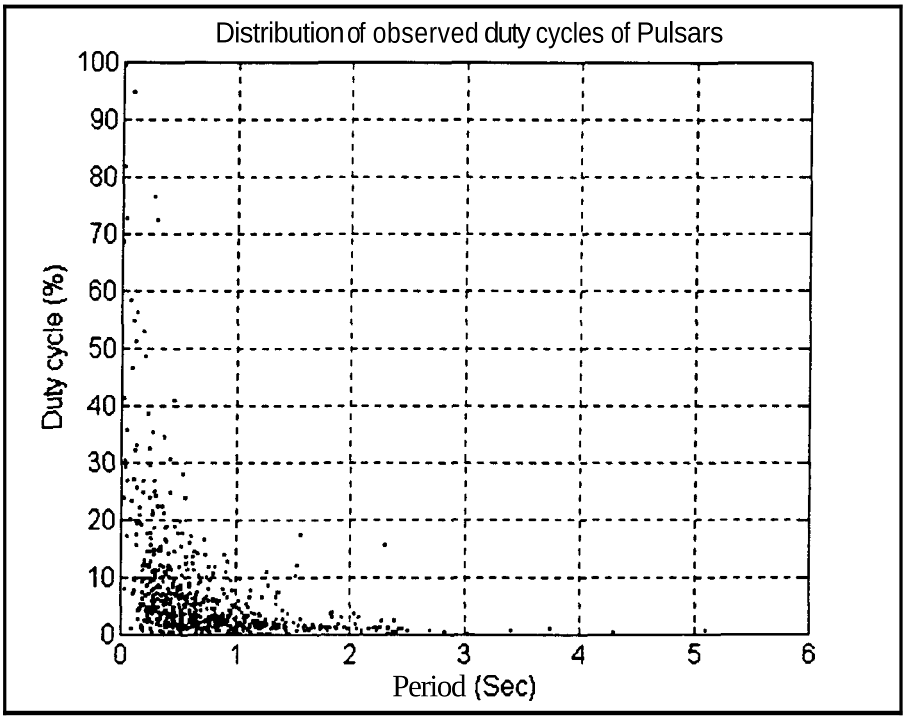

Fig 2.4

The data recording rates may become prohibitively large for low DM pulsars and/or at high frequencies where the attainable time resolution and hence the finaal sampling rate is high. In such cases the signals can be gated before recording so that only the on-pulse regions are selected and those during off-pulse regions are rejected. Looking at figure (2.4) which represents the duty cycle of profiles from known pulsars, it is obvious that in most cases only about a few percentage of the period is occupied by the pulse

Thus pulse gating technique will reduce the data rate requirement substantially. Typically, the gating over a

$$R = \left[\frac{1.5W \, . N_{channels} \cdot N_{stokes}}{\tau_{preint} \, . \text{Period}} \right] \qquad (2.11)$$

time width of about 1.5 times the pulse width (plus a similar time section for interpulses, if any) is sufficient to obtain good time resolution on-pulse and reasonable estimates of off-pulse mean and rms values. Figure (2.5) shows the savings in data recording rate with gating for the highest and lowest operating frequencies of GMRT. This method is difficult in a practical sense, since it requires that the gating pulse is exactly synchronous with the arrival time of the pulses. For real time implementation, the apparent period of the pulsar needs to be calculated frequently based on the models of relative motion of the pulsar and the earth. This period is to be used to locate the exact epoch of the pulse arrival so that the gate is turned on and off synchronously with the pulse window. The data rate for recording with N_{stokes}, number of stokes parameters and $N_{channels}$, number of spectral channelsis then given by

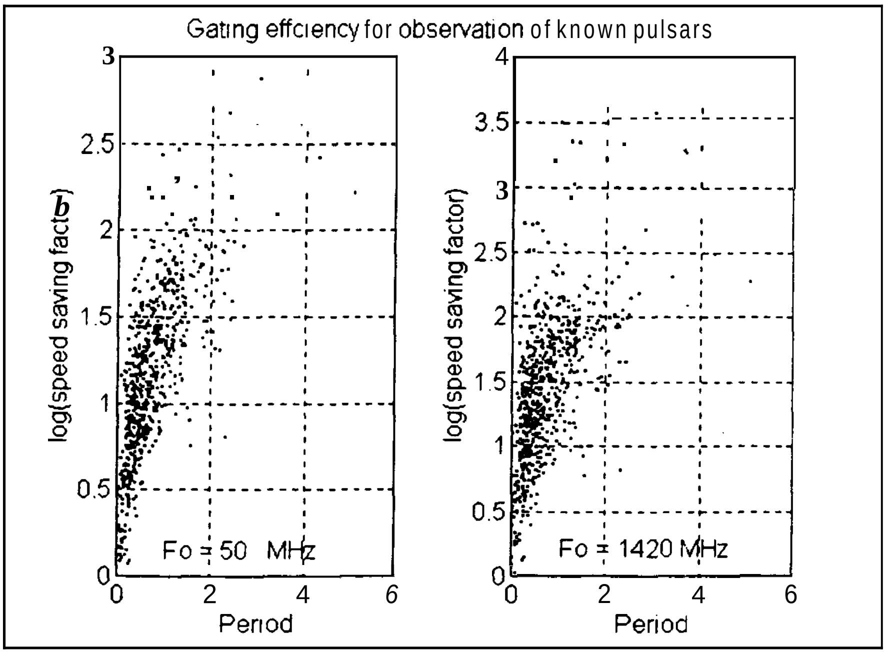

Fig. 2.5

Where $$\tau_{preint} = \sqrt{\left(\tau_{disp}{}^2 + \tau_{chan}{}^2\right)} \text{(= τopt, say, for further reference)} \qquad (2.12)$$

τ_{disp}, τ_{chan} are the pulse smearing widths due to dispersion and the finite channel bandwidth respectively. For the GMRT, 512 frequency channels are available, which can be collapsed to a lesser number of channels. It is usually sufficient to retain a few channels (say about 16) to help in proper estimation of flux, in the presence of interstellar scintillations which may produce intensity modulations across the band. It may also be necessary to reject some channels in the event of any interference being picked-up at some portion of the band. Keeping a few channels will avoid loosing the entire bandwidth during such events. Also, the uncertainty in the instantaneous polarization position angle measurement is about 3° at GMRT due to the finite data width of the Fourier spectra from the FFT modules. The resolution in frequency that may produce 3° smearing is about 16 channels across the optimal bandwidth assuming an uncertainty of about 50% in the kncwledge of the Faraday rotation due to the ionosphere. Polarization rotation with time due to parallactic angle correction also required to be corrected. The maximum recording rate for observations with the GMRT is calculated using the catalogue information of pulse-width (W) and DM for $N_{channels} = 16$ and $N_{stokes} = 4$ as shown in table(2.1).

Table (2.1)

Operating Frequency (MHz)	Total Optimal Bandwidth (MHz)	Maximum Data Recording Rate (Msamples/second)
50	1	0.05
150	4	0.3
233	8	0.6
327	8	0.9
610	16	2.7
1420	32	3

2.1.2.2. Studies of Average Profiles:

In these types of observations the aim is to study the distribution of energy within the pulse. Usually the signals are de-dispersed and corrected for Faraday rotation using the techniques mentioned in section (1.2.2.), and the frequency channels are collapsed after these corrections. Then, these pulses are folded over the period synchronously so as to enhance the signal to noise ratio, until the pulse shape becomes sufficiently clear(SNR of at least 10). This experiment requires that high time resolution is retained (limited only by dispersive smearing). Folding over a large number (Nfold) number of pulses will require a large dynamic range in digital representation of the integrated results. The output data rates will be very less, but a large memory may be required to store the temporary results during folding process. This temporary storage memory size may be reduced by gating the pulse as mentioned in the previous section. Gating over a width of 1.5 times the pulse width is usually sufficient to get the pulse structure and good estimates of off-pulse mean and rms values. The number of memory locations N_j required for such studies with above mentioned processing method (using optimal bandwidth) is given by:

$$
\begin{aligned}
N_m &= N_{stokes} \cdot N_{channels} \cdot N_{bins} \\
&= N_{stokes} \cdot N_{channels} \cdot \frac{1.5\ W}{\tau_{int}} \quad \text{where} \quad \tau_{int} = \tau_{opt}
\end{aligned}
\tag{2.13}
$$

Using $N_{channel} = 16$, and the catalogue information on pulse width W and DM for different pulsars, the maximum memory requirement at different observing frequencies of GMRT is given by table (2.2).

Table (2.2)

Operating Frequency (MHz)	Total Optimal Bandwidth (MHz)	Maximum Number of Memory Locations required (Kbytes)
50	1	15
150	8	86
233	16	167
327	32	263
610	8	524
1420	32	1.63×10^{3}

Also the total integration (No. of adjacent time samples added $N_{sampint}$ and total number of periods folded N_{folds}) required to achieve a signal-to-noise ratio of β times the noise rms σ, level can be calculated for these pulsars at various GMRT frequencies using equation (2.8) as

$$N_{int} = (N_{folds} . N_{sampint}) = \left[\frac{1}{N_{pol} . N_{channels} . \Delta_f}\right]\left[\frac{2K_b . \beta(T_{rx} + T_{sky})}{A_e . S_{avg}}\left(\frac{W}{P - W}\right)\right]^2 \qquad (2.14)$$

where the integration in each memory location is substituted for T_{obs}.

While the total integration for most weak pulsars would extend over several tens of minutes, it would be preferable to let the maximum on-line integration for only about a minute and record the results after this interval, and continue the rest of the integration during off-line processing, since the data size would have reduced considerably. This will be helpful in the presence of interference, where such blocks of data could be rejected, without loosing too much of data. Also, the receiver gain fluctuations may be compensated later suitably by taking integrated block once in such intervals. Then

$$N_{intmax} = \left[\frac{60 . N_{stokes} . N_{channels}}{T_{frame} . N_{m-min}}\right] \qquad (2.15)$$

where T_{frame} is the interval between successive spectra and $N_{m\text{-}min}$ is the minimum number of memory locarions required to hold one profile. Before integration the detected signal may have a mean power M and rms fluctuations of $\sigma_p \approx$ M so that, the total power is within M + $9\sigma_p$. But after integration, the exponential distribution of the detected power approaches a Gaussian distribution with a new mean $M_{int} = (N_{int}. M)$ and rms

$\sigma_{int} = (M\sqrt{N_{int}})$. The maximum value will then be mostly within $M(N_{int}+3\sqrt{N_{int}})$ and the corresponding bit-growth will be

$$\log_2 \left(N_{int} + 3\sqrt{N_{int}}\right) \quad \text{bits} \tag{2.16}$$

These values have been tabulated in table (2.3) for different frequencies of GMRT. As shown, the maximum growth in this case may be of the order of 21 bits

Also, the noise fluctuation have $\left(6\sqrt{N_{int}}\right)$ range around the accumulated mean count. While recording, it is sufficient to record limited bits that represent finally only a few bit variation across the window of $6\log_2\sqrt{N_{int}}$ around the value $(N_{int})\left(\sigma_p^2\right)$ where $\left(\sigma_p^2\right)$ is the pre-detection variance of the signal. This optimization gives a considerable reduction in the output bit length for small dynamic range (low SNR) cases. For strong pulsars, it is preferred to record the full width of the final data.

Table (2.3)

Operating Frequency (MHz)	Total Optimal Bandwidth (MHz)	Growth in word width
50	1	21
150	1	20
233	2	19
327	4	18
610	8	17
1420	32	15

It is also necessary to correct the variations in the period of the received signal by taking into account the phase change due to Doppler acceleration to make sure the folding is synchronous to the period. These integrated profiles can be recorded at regular intervals.

2.1.2.3. Dynamic Spectra Studies:

This study is mainly to monitor the apparent variation of the received pulse energy as a function of time and frequency. The area under the pulse window may be integrated to give the pulse energy, but in practice a few samples are produced across the pulse which can be later added with proper weightages (based on the intensity at different parts of the pulse), so as to avoid deterioration in SNR (Deshpande, 1987).

The pulses may be folded till a sufficient SNR is achieved, such that the integration time is still shorter than the de-correlation time for the process that causes the variations of interest (e.g. scintillation). Correction for Doppler acceleration may also be required in the case of binary pulsars. Gating may also be employed in cases when the period is too long.

In order to detect cases where the intensity fluctuations arise from ionospheric or interstellar scintillations, it may be necessary to retain all information across the frequency channels separately, but with data aligned after removing the dispersion delay gradient. While the intrinsic signal intensity of the pulsar is mostly constant across observed band, the scintillations produce systematic modulations in the power spectrum. Thus, such cases can be identified during off-line processing. Considering the choice of optimal bandwidth, the amount of adjacent sample integration required may be calculated as

$$\tau_{int} = \frac{1}{T_{frame}} \text{Maximum}\left[\frac{W}{5}, \tau_{opt}\right] \tag{2.17}$$

The data may be integrated upto this time constant and folded over an interval of about a minute before recording. Using this value of integration and equation (2.16), for various pulsars and operating frequencies of GMRT, the growth in number of bits, corresponding to the total integration for about a minute (tabulated in table (**2.4)**) is given by

$$N_{int} = \left[\frac{(60 \cdot \tau_{int})}{(\text{period} \cdot T_{frame})}\right] \tag{2.18}$$

Table (**2.4**)

Operating Frequency (MHz)	Total Optimal Bandwidth (MHz)	Growth in word width
50	**1**	**21**
150	**1**	**20**
233	**1**	**19**
327	**2**	**19**
610	**16**	**19**
1420	**32**	**19**

The number of memory locations required to store the temporary results during folding is calculated using equation **(2.13)** and the above integration time. (The maximum values at different GMRT frequencies are displayed in table (**2.5)).** Consecutive folded blocks of data, each of a time length of about a minute, can be recorded and processed off-line to study the variation of pulse energy with time.

Table (2.5)

Operating Frequency (MHz)	Total Optimal Bandwidth (MHz)	Maximum Number of Memory Locations required (Kbytes)
50	1	16
150	2	16
233	4	16
327	8	16
610	32	16
1420	32	16

2.1.2.4. Pulsar Timing Observations:

Clues to the age and evolution of pulsars, motion of pulsars in binary systems and readjustments in pulsar magnetosphere are reflected in the changes of observed pulse period (Manchester and Taylor, 1977). Pulsar timing experiment is devoted to the precise measurement of arrival time of the pulse (defined for example by the centroid of the pulse) and monitoring changes in the arrival time. Most of these changes that are systematic, may be related to the events taking place at the pulsar. These observations would require a very precise and stable timing device much better than the pulsar itself. Typically this stability will have to be in the order of one part in 10^{10}. Oscillators based on atomic transitions of Rubidium. Cesium and Hydrogen MASER are used as standard frequency sources to derive the local oscillator, the sampling clock and such other signals employed in the receiver. Usually the receiver produces the power spectra of the pulsar signal at precise intervals (much shorter than the pulse width). The train of samples in the different frequencies channels are summed together after correcting their relative arrival time differences due to dispersive delay gradient. The samples may be folded over the pulse period until the reasonable signal to noise ratio is achieved. Fast sampling is preferred since good time resolution helps in better identification of the centroid of the pulse. Usually the adjacent sample integration is limited only by the optimal bandwidth consideration of section (1.2.1.). Changes in period due to acceleration of the pulsars relative to observer may be modeled and should be incorporated during the folding process. Using equation(2.14), the minimum number of folds for a given SNR may be calculated. Using equation (2.16), the total growth in number of bits representing the data can also be estimated. Pulse gating may be employed in case of long period pulsars to save memory space. The number of memory locations required in this case is same as for single pulse studies and data rate will be similar to that for average pulse profile.

2.1.2.5. Average Profile Measurements of Polarization Properties of Pulsar:

Pulsar studies which are aimed at constructing the geometry and topography of the radiating cone require long term average profiles of full polarization data together with its spectral distribution (J Rankin, 1990). Usually, the signals are retained with a time resolution that is decided by the instrumental uncertainty and finite polarization angle resolution that is decided by the optimal bandwidths chosen. Table (2.6) shows the polarization smearing at the operating frequencies of GMRT.

Table (2.6)

Operating Frequency (MHz)	Total Optimal Bandwidth (MHz)	Maximum Polarization Angle Smearing (degrees)
50	1	93
150	1	14
233	2	7
327	2	3
610	8	1.6
1420	16	0.2

Typical instrumental uncertainty in the polarization angle may be of the order of a degree, so a time resolution of a **milli-period** may be sufficient. The consecutive spectra may be integrated till this resolution is reached. Then, the adjacent sample integration time constant is given by

$$\tau_{int} = \mathrm{Min}\left[\tau_{int}\ ,\ \frac{\mathrm{Period}}{1000}\right] \qquad (2.19)$$

where $\tau_{int} \geq T_{frame}$.

The spectral information contained in the frequency channels are retained independently but known effects such as Faraday rotation, parallactic angle changes, and changes in the instrumental phase can be corrected on-line. The data may be folded over a period for a length of time that is shorter compared to the time scales of ionospheric perturbations, but at the same time enough to attain reasonable signal to noise ratio. Perturbations in the ionosphere results in changes of Faraday rotation and in turn may result in depolarization if the pulses are over folded. However, these perturbations are only a few percent of the total ionospheric rotation measure and can be ignored in some cases. Depending on the variability of the ionospheric RM and other factors mentioned in section (2.1.2.1), a few channels can be collapsed to reduce the data. Also, gross correction for RM can be given and the residual Faraday rotation can be observed with time. The processing required for these observations poses the maximum computational load, since it involves all operations such as de-dispersion, Doppler correction, folding, gating, integration in time and frequency besides the Faraday correction.

2.1.3. Signal Processing for using Pulsar as Probes of the ISM:

Pulsars are useful probes of the interstellar medium. They act as torches, which illuminate the interstellar screen from behind, and the illumination turns on and off periodically, so that the differences observed between the on and off durations can be related to several physical characteristics of the medium.

2.1.3.1 Measurement of RM/Magnetic Field Distribution in the Galaxy:

As mentioned in section (1.2.2.), rotation measure (RM) is described as the rate of rotation of linear polarization position angle of the radiation passing through the interstellar medium with reference to the change in the frequency of radiation. The change in the polarization angle is given by

$\theta = RM.\lambda^2$ *and* the apparent position angle estimated by an observer is

$$\tan^{-1}\left[\frac{U}{Q}\right] = \Psi + \theta \tag{2.20}$$

where Ψ is the original position angle before the Faraday rotation.

Where (Q+jU) is the complex linearly polarized power at the given frequencies f_1 and f_2 represented by Stokes parameter Q and U at a given wavelength λ .

Also, the RM is related to the total electron content and the magnetic field in the line of sight of the observer as (Kraus, 1966)

$$RM = K_1 \int_0^L N_e \, B\cos\Phi \, dl \tag{2.21}$$

where N_e is the mean column density of electrons along the line of sight, $B\cos\phi$ is the component of magnetic field projected on the line of sight and K_1 is proportionality constant and dl is the incremental distance along the line of sight.

The dispersion measure (DM) is given by

$$DM = K_2 \cdot \int_0^L N_e \, .dl \tag{2.22}$$

where K_2 is another proportionality constant.

The DM can be obtained independently by observing the delay gradients in arrival times of pulses at different frequency channels in the spectra. Thus, if the electron column density in the interstellar medium can be assumed to be uniform, then DM and RM can be related to obtain the line-of-sight component of magnetic field using the relation (Manchester & Taylor, 1977).

$$\langle B.Cos(\Phi)\rangle = \frac{K_2 .RM}{K_1 DM} = 1.232\,\frac{\text{RM}}{\text{DM}} \tag{2.23}$$

The extent of Faraday rotation is a physically measurable quantity and can be found by taking the difference of the linear polarization angle observed at different frequencies, as follows:

$$\Delta\theta = \tan^{-1}\left[\frac{U}{Q}\right]_{\lambda_1} - \tan^{-1}\left[\frac{U}{Q}\right]_{\lambda_2}$$

$$\Delta\theta \cong \text{RM}\left(\lambda_1^{\,2} - \lambda_2^{\,2}\right) \tag{2.24}$$

$$\text{Hence} \qquad \text{RM} = \left(\frac{\tan^{-1}\left[\frac{U}{Q}\right]_{\lambda_1} - \tan^{-1}\left[\frac{U}{Q}\right]_{\lambda_2}}{\left(\lambda_1^{\,2} - \lambda_2^{\,2}\right)} \right)$$

The data consists of samples representing linear polarization of two widely separated frequencies. It will be preferred to observe the source simultaneously at two frequencies apart by a factor is greater than two or so, since the sensitivity of RM detection is proportional to f^3. With the GMRT, it will be possible to observe at any of its two frequencies simultaneously, by splitting the array into two portions, each portion operating at a different frequency. However, measurements at suitably closely spaced frequencies may also be needed to resolve the possible uncertainty of integral π rotations. The available band-width at most frequencies may provide such measurements. The data can be folded over its period for several minutes to average out the ambiguities in polarization angle measurement arising from ionospheric RM changes. Also, known effects which affect the polarization angle such as instrumental phase offsets and parallactic angle which may change with time must be modeled and removed from the data before folding. Since the polarization angle sweeps through the pulse as a function of the longitude, the integration of adjacent samples must be kept at minimum, to avoid polarization smearing within the pulse. Pulse gating technique may be employed to reduce memory requirements as discussed earlier. By retaining frequency channels of the spectra without combining (with only correction for dispersive delay gradient), the fit for RM value measured at widely spaced frequencies can be improved, by fitting the rotation of the polarization angle within the observing band for RM. Rapid changes occur in the RM due to travelling ionospheric disturbances, over time scales of a few seconds, and spatial scale sizes of few hundred meters to a few hundred kilometers (Spoelstra et. al., 1984). With the GMRT, these moving perturbations may smudge the polarization information, since antennas of the array extend over 20kms. The summation of the array output is expected to cause depolarization of the order of a few degrees in such cases. With pulse gating of about 1.5 times the pulse width, the maximum memory to contain all frequency channels, 4 Stokes parameters and time frames during the folding process is same as that of average profile polarization studies.

In case of very strong pulsars, the RM can be estimated in relatively much shorter time scales of a few seconds. In such cases, the estimate of RM obtained may even be fed back to the system as to adaptively correct for significant RM changes, while observing these pulsars (Ramkumar, 1991).

2.1.3.2 HI line studies:

Pulsar radiation may be absorbed by intervening interstellar medium. This absorption is mostly at characteristic frequencies, decided by the elements that constitute the ISM. The absorption is visible as a narrow dip in the spectrum when signal spectra of the on-pulse interval are compared with the off-pulse interval. The difference in the power spectra of the on-pulse and off-pulse regions will show characteristic frequencies of the absorbing clouds thereby revealing the cloud structure and composition (Manchester and Taylor, 1977). This technique is employed in detecting HI distribution in the interstellar gas. These observations require that the on-pulse and off-pulse regions to be kept separate. The pulses can be de-dispersed within the observed band but the frequencies must be kept independent. Only a few independent spectra are required in the on-pulse and off-pulse regions allowing averaging of data over wide-enough ranges of longitudes. Further, the pulses may also be folded on-line for several minutes to gain reasonable signal to noise ratio. These optimizations are usually enough to reduce memory size and data rates to a manageable extent. Folded profiles may be recorded with intervals of about a minute, and averaged further during off-line processing. Generally a machine designed to handle the types of signal processing tasks mentioned hitherto will also be powerful enough to handle these observations. Typically the data may be integrated and recorded once every few minutes.

2.1.4 Deriving a Common Processing Algorithm:

All the operations in the above processing modes can be classified into three basic types: Additions, Multiplications and data Indexing. For example, the basic operation on data samples during integration, pulse folding, channel collapse, pulse gating is addition of suitable data points. In Faraday correction, the main operation on the data is multiplication of data with suitable correction factors. During the process of de-dispersion, Doppler correction, gating, folding and integrating, locating the proper position in the time-frequency matrix where a datum needs to be added, coined as indexing, assumes prime importance. A careful analysis of the algorithms was done to identify the redundanciesisimilarities in these operations, which can be removediexploited to improve the speed and can be used to design a single framework to handle various jobs mentioned above. A step by step development of a combined algorithm to perform all of the above operations is presented below. It is important to note that these operations are common to all Stokes parameters, only the correction factors may be different.

a) De-dispersion (channel alignment):

As mentioned in section (1.2.1.), the data of different frequency channels of consecutive spectra can be written in a memory in the form of a time-frequency matrix with the profile in each channel being located in a N_{bins}, number of memory locations. Consecutive spectra are produced at time intervals of T_{frame}. This matrix of raw data will have a dispersive delay gradient as shown in figure (1.2), and the offsets in the arrival time of each frequency channel, relative to the highest frequency channel can be calculated in terms of the sampling interval as

$$\Delta N_{(f)} = \frac{\Delta\tau}{T_{frame}} = \frac{4150 \cdot DM}{T_{frame}} \cdot \left[\frac{1}{f^2} - \frac{1}{f_{high}{}^2} \right] \tag{2.25}$$

For a sample of a given frequency channel and time frame, this offset can be subtracted from the time index and the data can be written into the memory location addressed by the modified time index. This way, as the data of different frequencies get written into the memory, the dispersive delay is automatically compensated. When the data is read out sequentially from the memory, it will correspond to aligned frequency spectra. If the profile of one frequency channel and one period fits into N_{samp}, then for a given sample of i^{th} time frame and k^{th} frequency channel, the linear memory address is given by

$$A_{(i,k)} = (K \cdot N_{samp}) + REM\left(\frac{i - \Delta N_f}{N_{samp}}\right) \tag{2.26}$$

where the first term $(K . N_{samp})$ forms the base address (B_k) of the corresponding frequency channels and the REM operator extracts the remainder of the operand ratio. The layout of the memory for such a scheme is shown in figure (2.6) for two channels.The remainder in the second term indicates that if the index goes beyond the upper or lower boundaries of N_{bins}, it is wound back into the N_{bin} space, as though the time-frequency matrix were on a cylinder. This is useful for pulse folding, where in the next pulse can be overlapped over the current profile. However, for single pulse studies this wind back is avoided. Instead, the relative delay between channels is compensated by just skipping the corresponding number of samples in respective channels initially. For subsequent pulses, a fresh portion of the memory will be used and all address pointers will be shifted by a suitable offset, corresponding to the beginning of the old and new profiles in the memory. Since the profile in different channels may complete a period at different times based on the dispersive delay gradient, the offset addition is also to be performed as and when an individual channel crosses a period

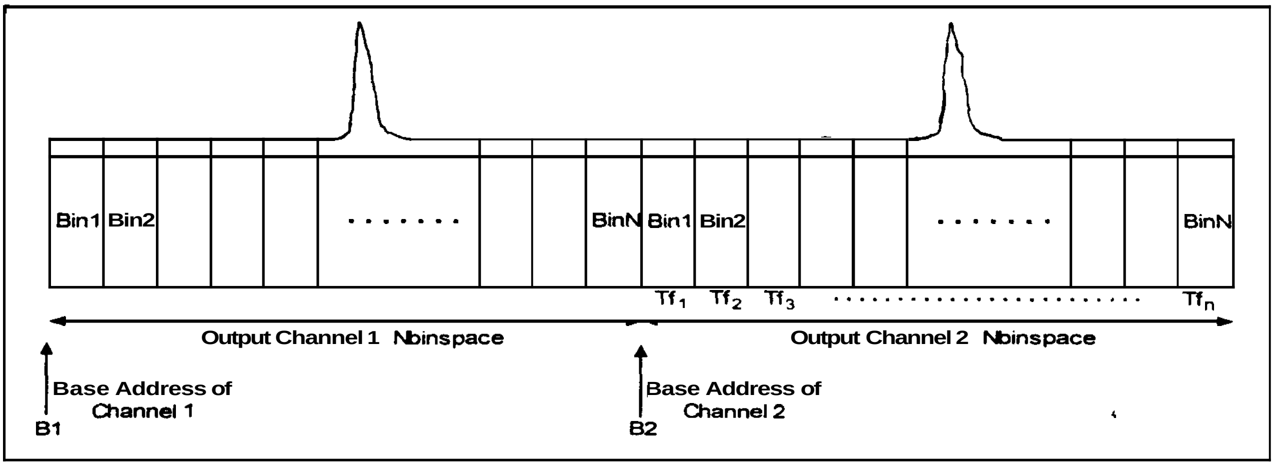

Fig 2.6 Memory layout for hosting profiles of Dedispersed Output Frequency Channels

b) Frequency integration (channel collapse) + Folding:

Let there be N_{chin} frequency channels and for the moment, let it be required that all the channels are to be collapsed together after appropriate dispersion delay compensation. As a new data of a particular frequency channel and time frame arrives, the previous sum at the target location is retrieved from the memory, added to the new sample and the result is written back. This can be achieved by simply ignoring the first term in equation (2.26), so that the profiles of all frequency channels get added to the set of memory locations assigned for the profile of only one channel say, channel K=0, such that $B_k=0$ for all k. In general, it is not necessary that all the channels be collapsed into a single profile. If N_{chout} is the number of output frequency channels, then the index calculation needs a slight modification. The memory can be split into N_{chout} banks of N_{bins} each. It is reasonable to let N_{chin} be a binary multiple of N_{chout}. Then, there are M channels to be collapsed to form each output channel, and the index is given by

$$A_{(i,j)} = \left(j \cdot N_{samp}\right) + \text{REM}\left(\frac{i - \Delta N_f}{N_{samp}}\right)\Bigg|_{i=1:Nsamp,\ j=1:Nchout} \tag{2.27}$$

Thus the index is jammed to contain same value for corresponding time samples of all channels that fall within each bank. As mentioned in the previous section, a suitable address offset may be added to the final address of equation (2.27), to handle the data for single pulse studies. In this case, the address offsets are also jammed to a appropriate values for all channels that fall within each bank, but they are added, to equation (2.27) at appropriate times when the pulse in a particular channel crosses one period. After the samples of all channels switch over to the alternate memory block, the first block may be recorded.

c) Pulse Folding :

For the purpose of illustrating this method, folding of pulse profiles on a single frequency channel is considered initially. If the pulsar has a period P and the interval between consecutive frequency spectra is T_{frame}, then the number of time samples within one period is given by:

$$N_{samp} = \left(\frac{P}{T_{frame}} \right) \tag{2.28}$$

This value, in general is a real number, consisting of a fractional and integer part. Since the number of memory location that have to contain the profile has to be an integer, the fractional part is the residual time width that has to be condensed by time-interpolation. If N_{bins} is the number of memory locations (equal to integer part of Nsamp at present), the residue per sample is then given by

$$\Delta_R = \frac{(N_{samp} - N_{bins})}{N_{samp}} = \frac{FRAC(N_{samp})}{N_{samp}} \tag{2.29}$$

where the function FRAC(X) is to extract the fractional part of X.

If the address pointer can be a real number, then the increment in memory address for every sample can be less than unity, given by

$$Ph_{inc} = (1 - \Delta_R) \tag{2.30}$$

Only the integer part of the address pointer can be used to access the memory. At the end of the first period, the address pointer will fall short of N_{bins} by an amount equal to $FRAC(N_{samp})$. This difference accumulates in a direction opposite to phase error created by fitting the profiles into N_{bins}. The error is compensated by shift of one bin in an appropriate direction every time the accumulation reaches half a sampling interval, i.e., $\frac{T_{frame}}{0.5}$, thus keeping the error within $\pm\left(\frac{T_{frame}}{2}\right)$. Thus, the profiles being folded drift only by (t) half a bin and the time-smearing due to this error is limited to this extent. The successive profiles align exactly in phase with the first period once every L periods, given by

$$L = \frac{2 \cdot T_{frame}}{FRAC(N_{samp})} \tag{2.31}$$

In general, when there are multiple frequency channels, this phase increment is common to all frequency channels. For a given output frequency channel j and time frame i+1, the absolute memory address can be calculated from the address A(i,j) of the previous time frame i, as follows:

$$Adrs_{(i+1,j)} = INT(A_{(i+1,j)}) = INT(Ph_{inc} + A_{(i,j)}) \tag{2.32}$$

where the function INT(X) to extract the integer part of X.

d) Doppler correction:

The apparent period at a given epoch may be different from the true period of the pulsar, caused by Doppler shifts in the observed pulse frequency due to relative motions of the Earth and the Pulsar. As mentioned in section (1.3.1), many standard algorithms and software routines already exist for calculation of the apparent period at a given epoch. The correction of interest is to compress or expand the profile before folding in a direction opposite to that caused by Doppler effect so as to match the length of the profile at the beginning of the observation. This is required to ensure that every new profile is in phase with the old ones as folding progresses, to avoid time-smearing of the features within the folded pulse. Once the new period P_{new} is known, the compression or expansion can be implemented by changing the phase increment Ph_{inc} value suitably. Let the ratio H be given by

$$H = \frac{P_{new}}{P_{old}} = \frac{N_{samp\_new}}{N_{samp\_old}} \tag{2.33}$$

Then the new phase increment Ph_{inc_new} is given by

$$Ph_{inc\_new} = \left\lceil \frac{Ph_{inc\_old}}{H} \right\rceil \tag{2.34}$$

Since the period changes gradually, the new phase increment value needs to be calculated frequently so that the period changes are tracked with fine resolution and the residual time-smearing is less than the time resolution of the pulse profile. The rate of change is small in most of the factors contributing to relative motion except in the cases of tight binary pulsars, where the orbital period may be as short as about an hour. Considering such a case (with a circular orbit), the interval between updates in Ph_{inc} values may be as small as 1 second so as to ensure that the period error does not cross one sample interval at the input time frame. This is not too fast for the hardware to handle. Since the residual phase error accumulates, the Ph_{inc} value should be over-corrected by half a bin so as to ensure that the mean error is always zero.

e) Successive Sample Integration:

In many cases when the pulse width is large and one can afford to have coarse resolution in the profile, it is preferred to sum up adjacent spectra, so as to reduce the data rate and memory size. For example, the residual dispersive delay within the channels after incoherent de-dispersion may spread over several time frames at lower frequencies. The information smeared within this interval, so one may as well integrate these samples, retaining only about 2 independent samples within each dispersive smearing interval. Integration can also be viewed as another form of compression or expansion of the pulse, except that it does not change with time. This can be implemented in a simple manner by just changing the number of bins into which the profile has to be fit into, i.e., N_{bins}. The new value of N_{bins} will then be given by

$$N_{bins} = \frac{N_{samp}}{N_{int}} = \frac{1}{N_{int}}\left[\frac{P}{T_{frame}}\right] \tag{2.35}$$

In case of using integration, N_{bins} **will** have to be used instead of N_{samp}, for all calculations of earlier functions, so as to scale the various parameters accordingly.

9 Pulse gating:

This option is to be used when a large number of samples fall within a period, so as to save memory space and data recording rate. The window covers the pulse, but the dispersive delay across channels results in the positions of the window get shifted to different time frames in different channels. Once the window width is decided, it is required to lock the window onto the actual received pulses in real-time. In the beginning of an observation, the window may be set at any arbitrary position. The beginning and ending positions of the window in units of pulse phase can be determined for each channel. Then, the folding logic can be provided with a comparator such that, as the folding progresses, the comparator indicates whether the current phase is within the window region. During the window region, the integer part of the phase which is extracted to address the memory location and subsequently the phase gets incremented. However, during the off-window phase, the phase increment proceeds, but the integer part is jammed to a constant value, so that all the time frames falling external to the window get added onto a single bin. This bin can be later ignored. By this method, full resolution can be provided for data within the window, and only one location is sacrificed for every channel, to integrate all the data external to the window.

To begin with, the entire profile may be folded without gating, enough to reach a reasonable signal to noise ratio, the resultant profile may be inspected to detect a pulse. The position of the de-dispersed pulse may then be aligned to appear at the beginning of the profile, by adding constant offsets to the current phase pointers of all channels. Then the window width may be reduced to allow only the on-pulse region. This adaptive phase-locking of the window can be automated under software control and within a few iterations, it must be possible to locate and lock the on-pulse region to the gate window. In case of pulsars with inter-pulses, multiple windows will have to be setup.

It is clear from the discussion above that all operations can be clubbed together, by simply manipulating the index pointing to the current phase-bin.

g) Faraday de-rotation:

As discussed in section (1.2.2.), the Faraday rotation causes the linear polarization angle to sweep at a non-linear rate across the different frequency channels of the band. It is sufficient to correct the rotation in different frequencies relative to that in the highest frequency channel. The differences in the rotation at individual channels are calculated using equation **(1.12).** Faraday correction is a simple phase rotation to be performed on the Stokes parameters Q and U. It is preferred to supply the correction angle in rectangular co-ordinates and multiply the terms

if $\Delta\phi$ is correction angle, then let $(\Phi_R + j\Phi_I) = (\mathrm{Cos}\Delta_\Phi + j\mathrm{Sin}\Delta_\Phi)$

The correction is performed by the multiplication : $(Q + jU)(\Phi_R + j\Phi_I)$

The resolution of digital representation of Φ_R and Φ_I need be only slightly better than that of the data in Q and U.

2.2 Engineering Considerations for System Design:

An instrument capable of handling all the above signal processing tasks in real time for the various types of observations mentioned in the previous chapter is bound to be complex. During the design of such a system many practical limitations in implementation aspects will decide the performance of the instrument.

In general, the design should maximize the following performance factors:

1. Speed of operation.
2. Flexibility to choose the appropriate signal processing algorithm for different types of pulsar observations.
3. Ease in reproduction, upgradation and diagnostics of the system.
4. Long term reliability.
5. User friendly controls and messages.

At the same time the design should minimize the following factors:

1. Power consumptions.
2. Quantization error.
3. Development, reproduction and trouble shooting time.
4. Physical size of the instrument.
5. Cost of implementation.
6. Electrical Interference from this instrumentation.

What can be achieved for most of these factors depends on the availability of technology and market trends. The state of the art and cost may change significantly from the time of freezing the design to the full implementation of the system. To gain advantage of these market trends, components are purchased in several installments. Keeping the above points in view the discussions that follow present the strategies used in the design of a pulsar signal processing instrument for the GMRT.

2.2.1 High Speed Discrete Logic Design Optimizations:

At the time of this design a survey of components available in technologies like ECL, TTL, CMOS was conducted and their parameters such as operating speed, power consumption, density, functional, etc., were compared. This study showed that CMOS devices with TTL level interface (for example, FCT, HCT devices) were the best choice for high density and low power considerations (Pericom High Performance CMOS/BiCMOS Data Book - 1995). However, the propagation delay of these devices range between 5-15 nano seconds, thereby limiting their reliable speed of operations to about 20 MHz. Also, beyond this speed, PCB layouts demands extreme care in terms of limiting the transmission line lengths, component lead length and compact placement of components & tracks to avoid signal deterioration from transmission losses and cross-talk. Also, the cost of the devices increases so rapidly with speed that it is cheaper and more reliable to build two parallel system each working for 16 MHz band rather than building a single system for the entire 32 MHz band. Considering the above factors, the pulsar receiver design is implemented as two identical halves, each catering to half the GMRT bandwidth. With this architecture, the design can be realized comfortably with TTL/CMOS logic, while ECL devices have been used only in transmission links between the pulsar receiver and other systems of the telescope, where high-speed communication is required over distances of a few meters.

Many computations in the processing hardware require complex multiplication and addition of numbers at high-speed, typically, once in every **64** nano seconds. As an example, consider a computation $Y = A' + B^2$, where A. B are represented by 8 bits each.

To realize this calculation in digital logic one will require two 8-bit multipliers and 16-bit adder, all of which have to operate fast enough so that the delay between presentation of A, B and output Y is less than 64 nano seconds. Devices that work for this speed are extremely rare and costly. Two alternate implementations are possible:

2.2.1.1 Look-Up Table Implementations:

This is an elegant shortcut method using high-speed PROMs, which work as Look-up tables.

The terms A and B can be presented as address inputs to the PROM. For any combination of A and B, a unique address is formed to the PROM. The contents of the memory location at this address may be pre-programmed to have the result Y corresponding to those values of A and B. In this way, the entire computation is pre-calculated and frozen into the PROM for all combinations of A and B. Later this PROM can be used in the circuit to maximize the speed of computation and minimize the hardware requirement. Also, any other function of A and B which produces the result limited to the same size, can also be implemented by just reprogramming the EPROM to contain the results of the new function. However, the application of this technique is limited by the number of bits used for input representation. This is because the locations required to store the results for all valid combinations of, for example, X and Y is $2^{(n+m)}$, where X and Y are represented in 'n' and 'm' bits respectively. As the number of bits increase, the unavailability of standard memory modules and the associated costs makes it prohibitive to choose this as an alternate scheme. Typically, the largest EPROMs available with acess times in the range 35 to 20nsecs are of the organizations 64K x 16, 128K x 8 or so, costing about $30 to $60 per chip.

2.2.1.2. Pipelined Logic Implementation:

In many occasions, when the number of bits per data becomes large, it is preferred to perform the computation using dedicated logic block. In such cases, lower speed devices may be used and the steps involved be pipelined so as to achieve the required throughput speed.

Clearly, the bargain for lower speed devices drastically increases the number of pipelined stages, and hence the number of devices routing complexity, size and cost of the system.

2.2.2 High Speed Programmable Logic Design Optimizations:

In both cases mentioned above, off-the-shelf ICs, such as TTL and CMOS devices provide specific logic functions and cannot be modified to meet individual design requirements. Also, the number of devices required to implement a typical design is usually large and the connectivity between these devices becomes very complex, demanding densely routed, multi-layer PCBs. The design may have to be split into multiple boards and subsystems when the board sizes start becoming too large to have things fitted in a single board. In such cases data communication and control between these modules and subsystems becomes more complex and become susceptible to cross-talks and interference when the system has to work at high-speed. Obviously, the flexibility to modify or upgrade such a system is also highly restricted. Besides this, as the device count increases, the power consumption increases accordingly and distribution of power at constant voltage to all subsystems and modules become increasingly difficult. Also, depending on the type of devices required to implement all the sub-functions of the logic design, a mix and match of available logic function in the discrete IC families has to be carried out and a large inventory of these devices has to be maintained. In many cases such as that of the pulsar receiver, the number of each type of device required for the system implementation and spares will be less in comparison with the minimum order quantity set by the vendors in purchasing the components. One alternative approach is to design custom made ICs (application specific ICs or ASICs). But this approach involves a large expenditure towards setup charges of the design lab and the foundries to manufacture the chip in quantities required by us which usually may not be large enough to make this approach an economical one. Besides the development and testing cycle takes a long time (about an year) before the reproduction cycle can begin, and clearly, the flexibility to change or upgrade is practically lost.

In the Design of the pulsar machine, a new approach is used: the implementation of reprogrammable logic devices. At the time of this design, the field of programmable logic implementation is still in infancy but is growing very fast. The idea in programmable logic implementation is simple: the IC provided by the vendor is a CMOS chip containing a large number of gates and flip-flops and a large, programmable interconnection matrix between different groups of gates and flip-flops. Similar to programming an EPROM, these interconnections can be retained or broken as per user requirement to realize any combinatorial or sequential logic. Such programmed devices are erasable electrically or under UV light, and can then be reprogrammed to contain a different circuit. These devices in general are called PROGRAMMABLE LOGIC DEVICES (PLDs) and FIELD PROGRAMMABLE GATE ARRAYS (FPGAs). Some of the FPGA devices are SRAM based, and the device configuration is down loaded to the chip after it is powered on, under the control of a separate controller. These devices provide the flexibility of programming the circuitry into the FPGA on-the-fly, while the system is powered on, without having to remove the device from the board. At the time of this design, EPLDs were available with gate densities ranging from 600 to 8000 gates per chip, with about 32 to 700 flipflops, working at propagation delays in the range of 12 to 25 nsecs. When the circuit is to be designed and frozen into the EPLD becomes large, it is cumbersome to handle these designs manually. Most of the vendors provide CAD packages which aid in the design entry, Boolean minimization, logic synthesis, timing analysis, functional simulation, device routing and finally, programming the device.

With the help of the development system and a host of devices provided by the vendors, it is possible to implement complex designs into a single or a set of programmable devices. The greatest advantage is that the functional and electrical behavior of the circuit can be simulated and understood even before the actual circuit is frozen into PCBs. The applications of EPLDs is best suited to situations where the data rates are too high to be handled by either the look-up tables (limited by the access times of the PROMs/RAMs) and where the use of specialized microprocessors such as digital signal processing chips (DSP) is limited by speed. Also, in cases where the circuit has to do a single, dedicated job, use of many high speed DSP chips may prove to be an overkill, considering the cost and support logic required for these chips.

In the design of pulsar signal processing instrument, EPLDs of different densities and pin-outs have been used in the different stages of the machine, depending on the optimizations mentioned above. After considerable market study, the family of EPLDs and FPGAs from M/s ALTERA Corp., USA, has been chosen as the standard for the designs in this instrument (Altera Technical brief, 1996). Since the field of programmable logic devices itself is growing rapidly, availability of good technical support and information on these devices from local representatives has also been a key factor making the choice.

2.2.3 Microprocessor Based Design:

The application of look-up tables and PLDs is limited to the extent of a design of a high-speed dedicated logic design, implemented to perform a specific function. To change the functionality, the design itself may have to be reprogrammed. In many cases the new algorithm to be implemented may have a different number of significant bits to represent the data, it may have to work at a different speed and may involve different input and output terms. When the functionality changes routinely depending on the type of observation as mentioned in the previous chapter, it may be best to optimize the algorithm independently for each type of function so as to reap the best benefit of speed and memory space. In cases such as this, it is appropriate to use microprocessors for implementing the algorithms under the control of a program. Most microprocessor manufacturers provide development support software on PCs, where the algorithms can be implemented in Assembly or 'C' language and tested with simulated input vectors before down loading onto the target system. This makes the development cycle faster and more user-friendly.

There are many general purpose microprocessors available in the market, but only a few microprocessors have specialized architectures to suit DSP applications. In designs such as ours, it is preferable to use these architectures since they deliver more computing power for a given clock speed and simplify the program development process. Some of the prominent architectural advantages of the DSP chips available at the time of our design are elaborated below to provide an insight into the optimizations utilized during the system design.

- **Modified Harvard Architecture and Intelligent Cache System:**

Some DSP chips provide two separate sets of bus and bus address generators for program and data memory, as shown in figure (2.7). This helps in access of data required by a "currently" executing instruction while fetching simultaneously the next instruction from the program memory, thereby enhancing the overall speed of computation. In some DSP chips, an on-chip intelligent cache is provided. In such DSPs; data can be stored in both program and data memories besides storing the program in the program memory. Whenever an instruction requires data from the data memory, it will be 'accessed without the overhead of an additional memory access cycle. But if the instruction has to access a data stored in the program memory, an extra cycle is required, since both the code and data have to be read from the same memory. The intelligent cache controller handles this situation by sensing such instruction that may need extra cycles and caching them inside the chip, along with the partial address of the instruction as soon as such an instruction is encountered. Hence during the first encounter, an extra cycle will be spent in data access from program memory, but for subsequent execution of the same instruction, (say in a loop) the instruction will be delivered by the cache. The cache controller can identify the instruction to be sourced by comparing the next instruction's address with the partial address information stored on the cache. By this method, the chip will pose essential three parallel independent busses - two for data and one for code and will result in an enhancement communication of speed by a factor of three.

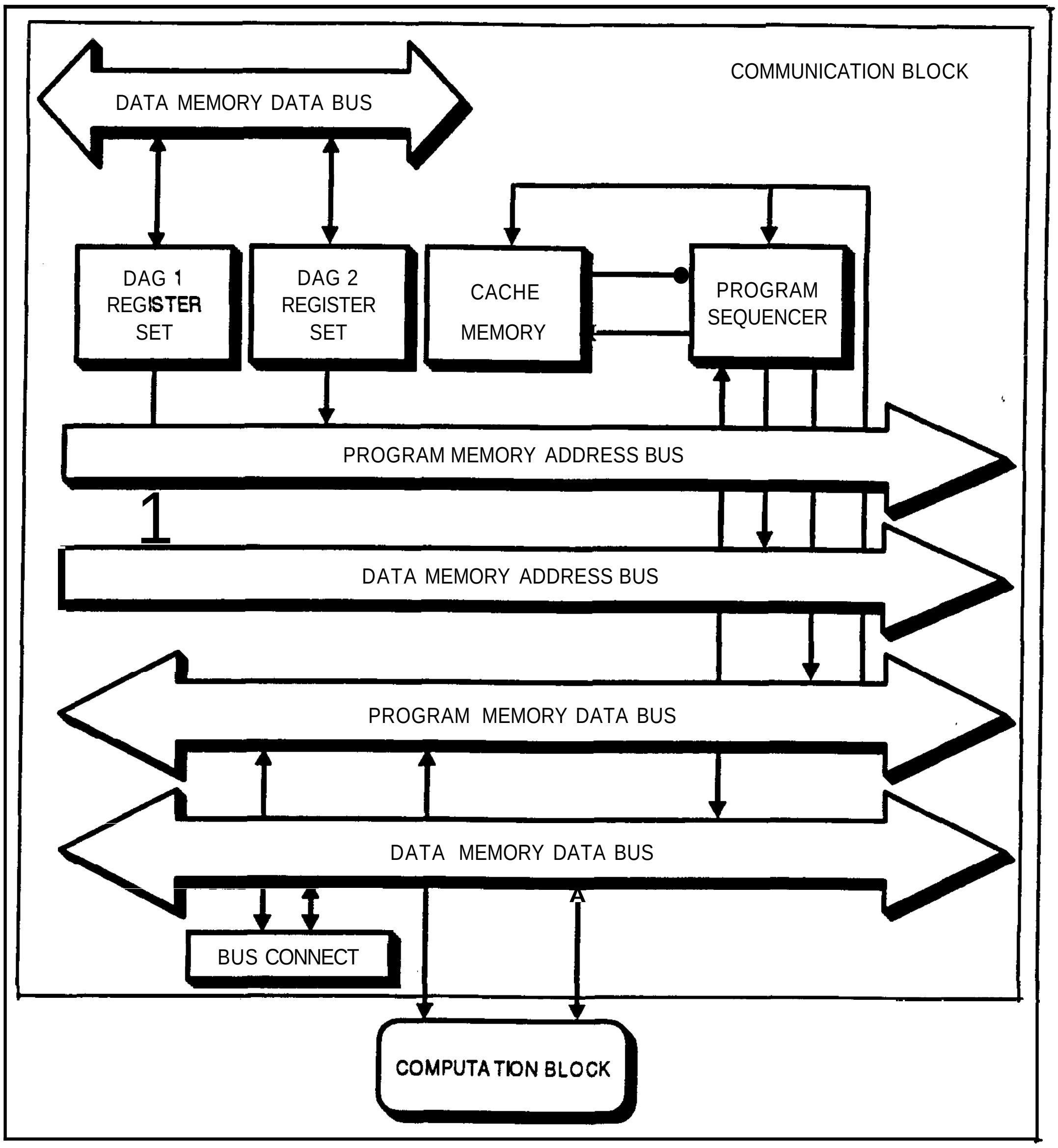

Fig 2.7 Address and Data bus Architecture of a modified Harvard DSP Chip

- Single **Cycle** Instruction Execution:

Most DSP chips have on-chip frequency multipliers and state sequencers built-in, so that all sequences of operations and pipelined execution of computations take place at much higher frequency internal to the chip, and all instructions supported by the chip will appear to consume only one cycle of the external clock supplied to the chip. Thus, a floating point add or multiply, a memory access or branch in the instruction flow will require only one clock cycle. This also simplifies program design for real-time applications. With this feature, the external interface design and clock circuits become simpler and also enhance the computational speed of the chip.

- **Wide Bus Architecture:**

Most of the latest DSP chips provide separate data and address buses, with the width of the address bus representing the full range of the direct addressing space accessible to the chip. Also the data busses are wide enough to fetch entire instruction codes and data in a single cycle. Even though this increases the number of buffers, etc., the overall interface design is simplified and allows single cycle memory transactions, thereby enhancing speed of instruction and data access.

- **Parallel Instruction Execution:**

Conventional microprocessors are provided with a pipelined architecture to fetch instructions, decode them, fetch their data, perform the computation and write the result back into the memory, popularly known as the Von Nueman architecture (as shown in figure (2.8a)). However, the DSP chips usually operate with parallel flow in these operations, controlled by separate blocks, as shown in figure (2.8b). An independent instruction handling block may control fetching, decoding of the next instruction, while another block simultaneously fetches data for execution of the previous instruction. Yet another block may control the execution of the current instruction in parallel with the above blocks. Some DSP chips provide parallelism in execution of instructions (operations such as multiplication, multiply and accumulate, logical/arithmetic shifts, addition /subtraction can all be done simultaneously on different data operands available in the DSP chip's internal registers). For example, the DSP chip from M/s.Analog Devices used in the present case, can do many concurrent operations within each loop in parallel, including loading, updating and checking the loop counter, within a single cycle of the clock supplied to the chip of say, typically **40** nano seconds, for example, as shown below:

```
LCNTR=RL, DO XXUNTILLCE, [loop Register value RL number of times)]
{
MR = MR + ( RA * RB) , [Accumulator = Accumulator + (operand1 * operand2)]
RY = RC - RD, [data register Y = data register C – data register D]
RZ = RC + RD, [data register Z = data register C + data register D]
RM ⇐ DATA MEMORY DATA, [register M = fetch data from data memory]
MODIFY DMA BY M8 , [modify data memory address pointer by M8 register value]
RN ⇒ PROGRAM MEMORY DATA, [store contentsof register N to program memory]
XX : MODIFY PMD ADDRESS BY MO; [modify program memory address pointer by MO register value]
}
```

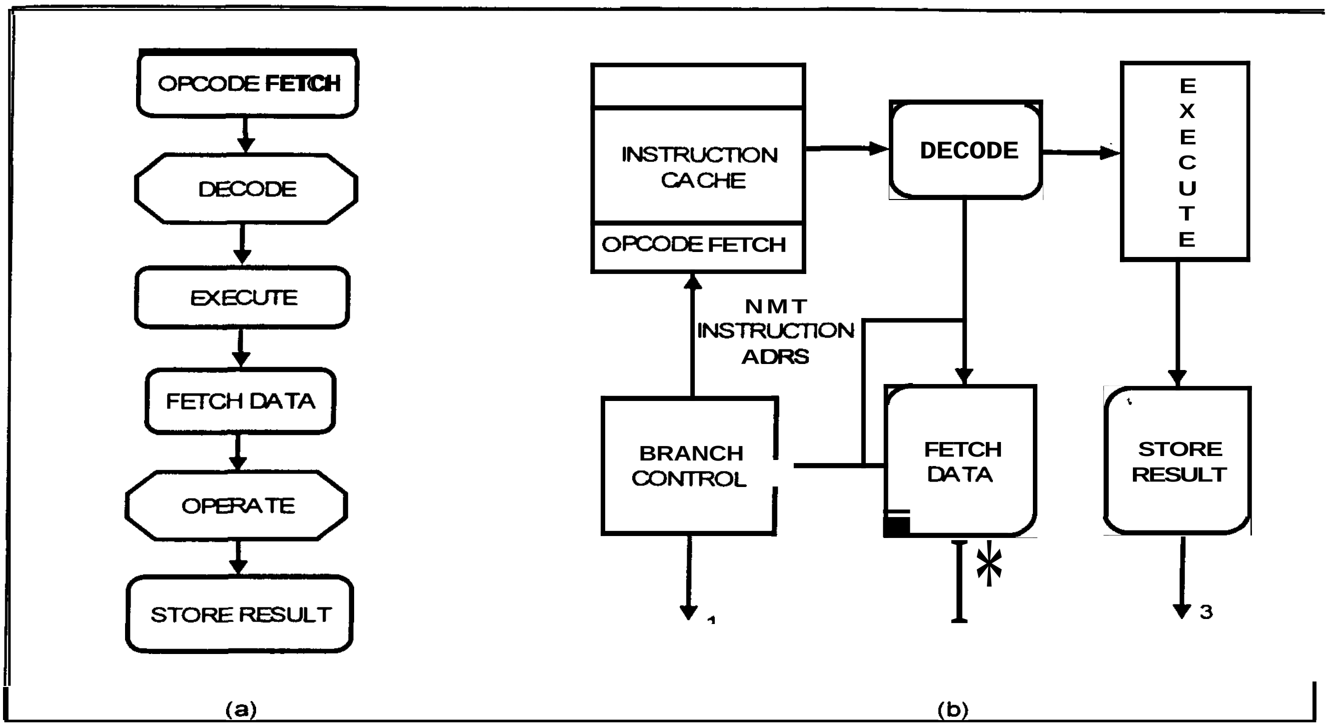

Fig 2.8 (a) The serial events in VonNueman architecture. (b) Parallel events in DSP architecture.

Even though this makes the programming a bit tricky, appropriate parallel coding of the programs can enhance the computing speed significantly.

- **Fast Context Switching:**

Many DSP architectures provide alternate sets of registers in the control and computing blocks mentioned above, so that the chip can keep working on one set normally, and when tasks are to be switched, it can simply leave the context of the previous task intact in the normal set of registers and switch over to the alternate set of registers to process the second task. This will isolate the transactions of the two tasks and facilitate rapid context switching in one clock cycle. Some DSPs also provide optional communication between the two sets of registers, so that, if required, one task may interact with the semaphores and data values of the second task purposefully.

- **Circular Buffer Addressing and Internal Interrupts:**

Many DSP applications, such as ours requires the use of several tables containing coefficients, data or such other information. These tables may have to be looked-up repeatedly, as and when required, with the index of the contents running through the table from beginning to end repeatedly. In conventional microprocessor chips, the starting address of the table needs to be reloaded every time the end of the table is reached, in order to start a fresh read out from the beginning of table. In DSP chips, an automatic roll-back facility is provided, such that the user specifies the start address and length of a table in the memory, and the address pointer automatically wraps back and forth if the index crosses any of the two boundary addresses of the table, so that the table serves as a circular memory. This architectural advantage brings about fast memory transactions by saving the extra computations required for index management. Some DSP chips even provide internally generated interrupts that occur when the boundaries of the circular buffers are crossed and the index is wrapped. This facility may be useful in automatically invoking an interrupt driven process to load new values into the list every time the list has been read out.

- **On-Chip Memory and Peripheral Controllers:**

Some DSP chips provide internal memory for both data and programs, in addition to peripheral controllers for serial and parallel communications, DMA controllers, timers, etc., for simplifying external interface and providing high throughput rates between memory and I/O devices and the registers of the chip. This feature is useful only in applications requiring relatively smaller data and program memories since on-chip memory are limited in depth.

- **Instruction and Data Stacks:**

Most DSP chips provide separate stacks and associated controllers to automatically stack the current instruction code, fetch address and contents of important data and control registers when events such as branches, exceptions and interrupts occur during the flow of the execution. This reduces programming overheads in saving contexts and enables nesting to multiple interrupts.

- **Zero Overhead Branching and Looping:**

In some DSP chips, the logic evaluating the test conditions for branching or looping will determine the test flags a few instructions earlier to the last instruction of the loop. This is possible since the instruction fetch, decode and execute operations are in a pipeline inside the chip. With this feature, the next instruction at the branched address or the loop address can be fetched and decoded in advance, so that no extra cycles are required during the branch. This feature can be used optimally to enhance the speed of execution, especially when several functions and subroutines are to be called repeatedly. At the time of this design, only a few companies were providing stand alone DSP chips (Markus and Anne, 1996). Among these, the ADSP-21020 from M/s. Analog Devices suits well for our current requirement. In addition to the these features, the availability of tools required for DSP program development, simulation and debugging in C/Assembly languages, have also played an important role in the selection of the chip.

2.2.4. High-Speed Memory Interfacing Considerations:

At some locations in the instrument, such as the DSP interfaces, data flows at 25 Mbits/second on wide address, data and control busses. Memory devices that need to be interfaced to the DSP chip are to be chosen such that their access time is as small as 10 to 15 nano seconds, to accommodate additional delays produced by address decoders and buffers. Buffers and line drivers pose a tradeoff between drive current and switching speed owing to the slew rate of individual devices and need to be carefully chosen. The buffer propagation delays will have to be minimized to relax the memory access time requirement, since the cost of memory increases very steeply with speed for a given density.

At many stages of the data flow, memories may be required to temporarily store and supply data to the next stage when the two stages may be using the data asynchronously. Two special types of memories, namely the DUAL-PORT RAM (DPRAM) and the FIRST-IN-FIRST-OUTMEMORY (FIFO) are found to be very useful in such applications.

In a DPRAM, two completely parallel independent ports are available, so that two subscribers (systems) can independently access any two locations inside the memory, simultaneously. The two ports are electrically independent and whenever both ports access the same loation the logic provided on the DPRAMs which gives a busy signal to the left or right port depending on which port is accessed later, so that the system requiring the service of that port can wait till the other port completes its transaction on that location and the busy signal is released by the DPRAM. As long as this condition is satisfied, the two ports can access the memory locations in any order and at any time. The DPRAMs are also equipped with an address-based interrupt generation logic. When the write cycle is performed on the left port at a specific address an interrupt flag is generated on the right port. This interrupt flag deactivates after the right port performs a read cycle on the same port. A similar mechanism is also provided for the interrupt generation from the right port to the left port. This is particularly useful to indicate that the certain block of data is filled by one of the ports so that it can be used by the logic connected to the other port without having to keep polling for such an event. This suits any application where **a** logic that delivers data, works at a different rate as compared to the speed of the logic which accepts the data.

In the absence of DPRAMs, logic will be implemented with the help of two memories and an elaborate set of tri-statable buffers or bus-multiplexers, which consume more space and interconnections on the PCB besides additional delays associated with these buffers. However, DPRAMs are limited in depth (typically about 8K X 16) and become very expensive at higher densities. In the design of this instrument, DPRAMs have been used only where the application of these devices is technically more reliable than that with the SRAM alternatives.

FIFOs are much like the DPRAMs, except for the fact that two separate address pointers, one for the each port are built into the chip. These ports are dedicated read-only and write-only ports. Also the read and the write address pointers increment sequentially on both sides and the address pointers increment automatically upon receiving read and write pulses on respective ports. FIFOs internal logic provides flags to indicate when the FIFO is "empty", "half full" and "full". FIFOs are particularly useful in applications that demand asynchronous, uni-directional transfer of data between two logic circuits in a sequential manner. For example, the input data may frequently arrive in a burst at some speed, while the storage system may read-out record from the FIFO at its own convenient speed. The only condition that is needed to prevent an overflow in the FIFO is that the average read-out rate must be greater than or equal to the average data input rate. The FIFOs simplify the interface by absorbing address generation function into the chip and 'by providing the above mentioned flags to indicate the extent of the data in the FIFO. Also, the FIFO depth can be changed without any alteration/additional logic since FIFOs of various depth are available in pin-compatible IC packages. However, FIFOs are to be used with careful interfacing, since they are sensitive to glitches on the control and power supply lines which may cause spurious increments in the read/write pointers.

2.2.5. Computer Based Control and Diagnostics:

As the circuit design becomes bigger and more complex, it can become increasingly difficult to provide user friendly, programmable , automated control and diagnostics facilities for regular health check for the system, which is essential since the instrument has to work continuously for days during observations such as pulsar surveys, etc. The options available are either to build a diagnostics module at each stage of the machine based on conventional micro-controller chips or dedicated logic. Both of these options require additional components and have to be custom built for each stage of the instrument. We have provided the diagnostics and setup controls using simple parallel I/O interfaces based on a standard PC-AT, using the ***Industry Standard Architecture (ISA)*** bus. The PC will program the 110 lines suitably to generate the control signals to setup the machine and supply diagnostic patterns, besides reading back the status and results from the instrument. Diagnostic software can then analyze the status/results to identify and report errors and possible locations and causes of these errors. The large base of software tools and hardware interfaces available off-the-shelf in the PC industry make it highly economical, easier and faster to standardize and implement the above features. Also, standard Ethernet interfaces available on a PC-AT allow the system to be hooked on to a remote machine, and be controlled from there. The PC may initially setup the entire machine to run in a user-defined mode. Then it supplies the data and controls the clock, so that the entire machine can be paused after every clock, to single step and check the outputs at each stage. On each clock the PC can supply a user defined data and read back the status and results from "diagnostic" points in the machine and log them into files. Later analysis programs can read the result files and compare them with expected results, locate errors, and their possible causes. Typically these errors may include spurious data, missing data, wrong address sequencing, open or short circuited loads, frequency dependent interferences in data/address/control signal paths, etc., which provide be a very useful picture to understand the behavior of the machine.

2.2.6. Interconnection of Components:

Devices capable of switching rapidly require effective de-coupling of "switching noise" in the power supply lines, besides short track lengths in the interconnections between the devices to avoid ground bounces and reflections along the tracks. Most of the control lines need proper termination at the receiving end in order to match the source and load impedances properly and ensure a ripple-free waveform'. Multi-layer PCB design for high speed is an art in its own right, and most of the PCB design packages support very effective set of rules in laying out the tracks of the PCB. The packages also provide tools to predict possible weakness in the layout, such as probable regions of cross-talk, insufficient track width to supply the required current, temperature distribution on the board, impedance mismatch due to long transmission lines, etc.. Even with the help of these packages, human interaction is imperative and pivotal in ensuring reliability and reproducibility.

Digital circuits may produce interference due to radiation of power at high frequencies, associated with the harmonic content of the switching signals. This switching noise may couple back either by radiation or through the ground lines that interconnect various instruments. The signals are to be shielded appropriately to avoid interference being picked-up by the telescope. Some of the standard steps that have been taken to ensure reliability are:

- Multi-layer PCBs have been designed with separate ground and power planes to provide effective de-coupling and shielding to reduce the susceptibility to external EMI. The PCB tracks have been provided curved edges at turning points, and the tracks passing in perpendicular directions have been placed in over-lapping layers to minimize cross-talk and radio frequency emissions. Also, the PCB track lengths have been kept at the bare minimum and have been terminated appropriately to minimize reflections.
- Grounds of various sub-systems of the instruments have been connected in star topology to minimize ground loops and cross-couplings between various ground points.
- The circuits are hosted in Aluminum cages and the cages in-turn have been hosted inside mild-steel instrument racks to provide shielding for any radiated interference. The steel rack is expected to produce absorption-loss while the Aluminum cage is to provide reflection loss. The entire instrument is to be housed in a Copper shielded room. Within this, the shielding is expected to be greater than 80 dB.
- The AC power supply lines are connected through commercial power-line filters to reduce intra-system interference due to power line transients.

- High-speed communication links connecting instruments at distances of several meters have been realized in the form of ECL-differential logic circuits so as to avoid common mode interference and reduce the slew-rate of the transceivers. Also, shielded twisted pair cables have been employed to carry these signals for shielding.

All the above mentioned optimizations in signal processing methods, circuit design and implementation techniques have been used in the realization of the pulsar instrumentation. In the following chapters, the instrument design, tests and results obtained for various segments of the pulsar signal processing system are elaborated.