CHAPTER THREE

PULSAR SEARCH PRE-PROCESSOR

This chapter describes the hardware and software details of a digital system built to handle on-line pre-processing for pulsar search. Implementation was taken up after the optimizations mentioned in the previous chapter were worked out. As a general philosophy in implementation, each of the major blocks of circuitry was functionally simulated by writing equivalent functions in C language and feeding the algorithms with random / deterministic data patterns before actual hardware realization. Table (3.1) gives the input base-band specifications of the Pulsar Search Pre-processor (PSP). In the presentation of the design, it is assumed that the front-end systems of GMRT and ORT described in chapter 1 are available. However, the results based on front - end systems available at the time of conducting tests are presented.

Table(3.1)

Base-band width	1,2,4,8 or 16 MHz per sub-band, 2 sub-bands per polarization, 2 polarizations.
Raw sampling rate and data representation	Nyquist rate, 1-bit sign, 3-it mag.
FFT output	256 channelcomplex spectrum per sub-band, polarization, generated every 512 time samples.
Array combiner output format	PA mode: 1-bit sign, 8 bit magnitude, IA mode: 8 bit unsigned magnitude each for real and imaginary parts of any frequency, polarization, sub-band. Two parallel outputs giving incoherent (detect & add) and phased array (predetection addition) combinations.
ıɛ masking at array combiner	Any set of antennas can be masked, two sub-bands can be used to operate at two different operating frequencies by masking different set of antennas in each sub-band.

Specifications of the front end system (Analog front-end, Sampler and FFT, Array combiner)

3.1. Specifications of PSP:

The block diagram shown in figure(3.1) outlines the different sections of the PSP. The specifications of PSP (table (3.2)) were derived after applying the optimizations mentioned in chapter two.

Table (3.2)

PSP input sequence and format	Sign bit dropped, 8-bit magnitude, 256 frequency channels arrive in time-multiplexed form, two parallel polarization channels per receiver, separate identical receivers for each sub-band.
Array modes supported	Software choice of Incoherent or Phased array mode, choice independent of polarimeter, which runs in parallel.
Pre-integration	Independent integration of consecutive time samples in each freq. channel , length of integration programmable from 256 μsecs to 4 msecs (common for all frequency channels).
Block integration	16 to 256 pre-integrated samples in each frequency channel.
Running mean Calculation	Simple moving average. time constant of 4 secs to 2 minutes using block integrated data (independent mean for each channel).
r lean subtraction	New running mean initiated for every block- integrated data sample and subtracted from every pre-integrated data sample independently for each frequency channel.
Gain calibration	Independent 8 bit scale factors for each frequency channel, equalization of gain with respect to the peak of the running mean spectrum.
Output quantization and bit packing	1-bit or 2-bit quantization (sign bit and any other magnitude bit) with bit-packing unto to fill 16 bit frames. Transmitted to DAS with synchronized write pulses.
Data acquisition	16 bit DAS, independent DAS for each sub-band. Sustained data rate upto 128 Kbytes per second. Temporary storage on hard disk, backup facility to port data on 4mm videotape.

The philosophy adopted in this design was aimed at reducing the size, complexity and cost of the machine while providing flexibility and operational simplicity. Two identical systems have been developed, each to handle one sub-band. The processor is equipped with hardware capability to receive 256 complex spectral channel data in each sub-band over 16 MHz base bandwidth, in time multiplexed format from the two polarization channels.

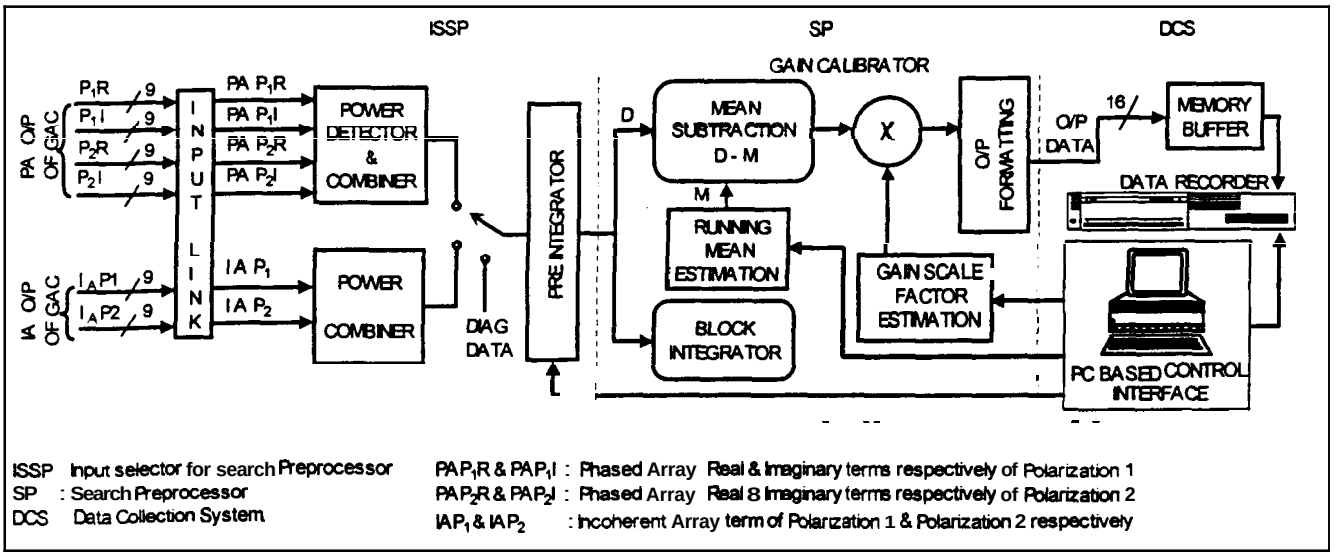


Fig. 3.1 Block Level Architecture of Pulsar-Search-Preprocessor.

3.2. System Architecture:

In this section, the implementation of different blocks of the PSP is described. The functioning of the entire circuit is described along with the data flow.

3.2.1. Input Link:

Since a new data sample appears once every **64** nsecs, all operations for a data sample are to be finished within this interval. This is not feasible given the propagation delays of conventional logic. To achieve this speed, the entire machine is split into various functional blocks that are pipelined such that the speed of logic used within each block is sufficient to finish the calculation of that block within **64** nsecs. The results of each block are then latched using edge-triggered registers and then propagated to the next block. By this method, only an initial pipeline delay of N clocks periods, corresponding to N registers in the chain, will be encountered. After this initial delay, a new data sample appears at the output of every stage including the final output once every 64 nsecs.

The PSP may be located at a distance from AC and PSP may co-exist along with many other digital systems for other observations and controls. This may render the link between the data links between the AC and PSP susceptible to local interference. The data to be transmitted from the AC to the PSP is sent at **16** Mbaud on each of the data lines to carry the signals corresponding to the real and imaginary parts of two polarizations of both sub bands of the incoherent and phased array outputs of the AC, in parallel. To ensure good data quality at the required speed, ECL differential links have been used between the **AC** and PSP. The data is converted from TTL to ECL levels at the **AC** outputs and from ECL back to TTL level at the PSP outputs. The signals are converted from ECL to TTL levels on the PSP side and latched into edge-triggered registers once every **64** nsecs, under the control of a **16MHz** system clock supplied from the AC. The data is transmitted on a twisted, shielded pair of **60** line flat cables.

3.2.2. Input Selector for Search Pre-processor (ISSP):

The input to this module is a set of consecutive spectra with a new spectrum appearing every 16

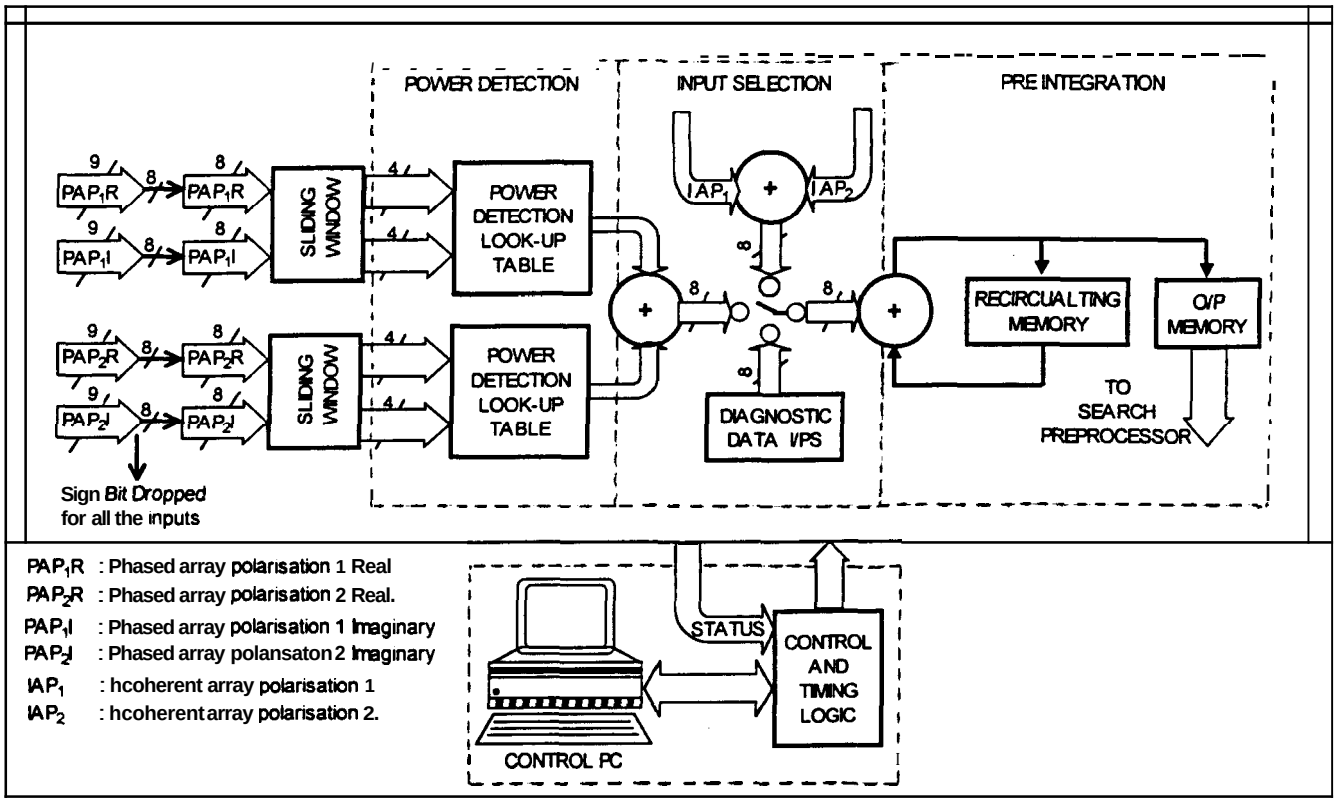


Fig. 3.2 Block Level ISSP module Architecture.

µsecs. Figure (3.2) shows a block diagram of the ISSP module. The ISSP module has two types of inputs: the incoherent array input (IA) and the Phased Array Input (PA). In the IA mode, the AC would have already detected the power and summed all the dish outputs to yield 8 bit numbers for each polarization. In the PA mode, the complex voltage samples of all the dishes would have been summed at the AC and they are available as complex numbers with the real and imaginary parts represented by 1-bit sign and 8 bit magnitude numbers at the ISSP input, represented in sign-magnitude form. One can choose 4 most significant bits of the magnitude that contain data within the 8 bit field, so that the real and imaginary parts are then 4-bits each (the sign bit is dropped, since it is not important for power calculation in the next stage). This simplifies computational logic, without significant degradation of the signal-to-noise. This is achieved with the help of a sliding-window logic based on a set of multiplexers, fed with staggered selection of bits as shown in figure (3.3). Depending on the number of dishes being combined,

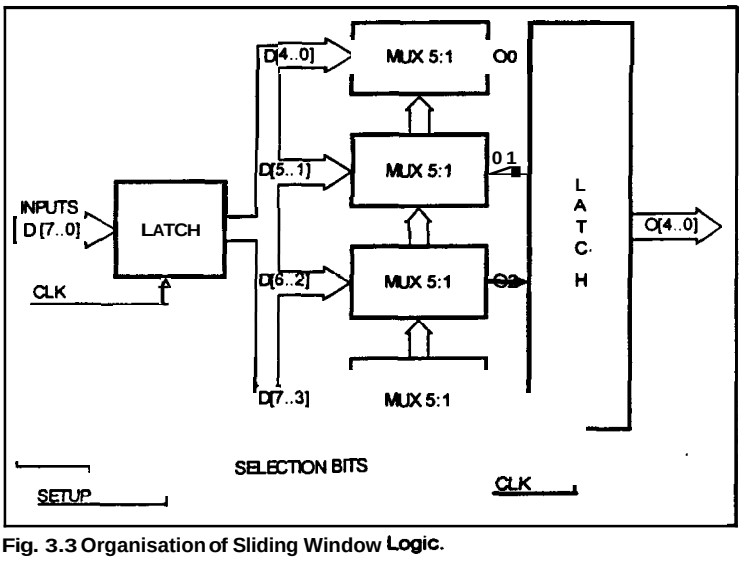


Fig. 3.3 Organisation of Sliding Window Logic.

the selection bits can be programmed just before an observation, so as to select the appropriate 4 consecutive bits in the 8-bit field. This logic is repeated for the real and imaginary terms of both the polarizations. An additional facility is provided to mask out inputs from any of the two polarization channels before an observation, to aid in diagnostics and fault isolation at the field. This entire selection logic has been implemented inside a single EPM5128 (25ns speed) EPLD chip, utilizing logic of about 128 gates & 48 flip-flops equivalent of (so many number) discrete chips.

3.2.2.1. Phased Array Power Detection:

The outputs of the sliding- window selection logic are passed onto a pair of look up tables consisting of a pair of 256 location, 8-bit PROMS (50ns access time). These lookup tables for each of the polarization channels contain pre-computed values of the "power" (rounded to 8 MSB bits) corresponding to the complex voltage (expressed with the real and imaginary parts) in the phased array channel. The PROM outputs are latched onto a set of registers (devices with about 10ns propagation delay).To improve the sensitivity for the detection of a pulsar, the "power" outputs of the two polarizations are added up. A three stage pipelined adder circuit collects the power terms from the lookup-PROMS of the PA channel and adds them. In parallel, it also sums the power terms corresponding to two polarizations of IA mode coming from the AC.The sum may grow to a maximum of 9 bits, and is, however, quantized to 8 bits, to match the standard width of available digital devices in the market.Further, a multiplexer circuit selects one of the following three sources of data for the next processing stages: a) total power from the PA channel, b) total power from IA channel or c) external diagnostic data source. The external source is useful for any field diagnostics in fault isolation, wherein a known pattern can be injected and the functioning of the later blocks can be checked. The selection of any of these inputs is programmable under software control, during the setup cycle of the machine just before an observation. This entire summing and selection logic is housed in a single 5128 EPLD using 40 equivalent gates and 48 flip-flops.

3.2.2.2. Pre-integration:

The samples representing the spectral power of consecutive frequency channels appear ,once every 64 nsecs as input to this stage. This rate can be reduced by integrating adjacent time samples of the respective channels, while retaining enough time resolution within the profile for all possible periods that would be searched in the survey. For this purpose, pre-integration intervals between 256 μsecs to about 4 msecs per time frame are found suitable. This would mean an integration 16 to 256 consecutive time frames in each channel. Since the data on each frequency channel arrives in a time-multiplexed fashion, the data is to be stored temporarily in a buffer with separate locations for the spectral channels.The accumulated sum of the corresponding channel is fetched from this buffer to the adder whenever a new sample of a particular channel arrives. As shown in figure (3.4), this " pre-integration " logic is implemented with the help of a pipeline adder, a circulating memory using a pair of dual-port RAMs (DPRAMS) and an output buffer memory made of 2 DPRAMS, in addition to an address generation and control logic. One input to the adder comes from the input selector EPLD, and the other input arrives from the recirculating DPRAM, whose read-address is initially

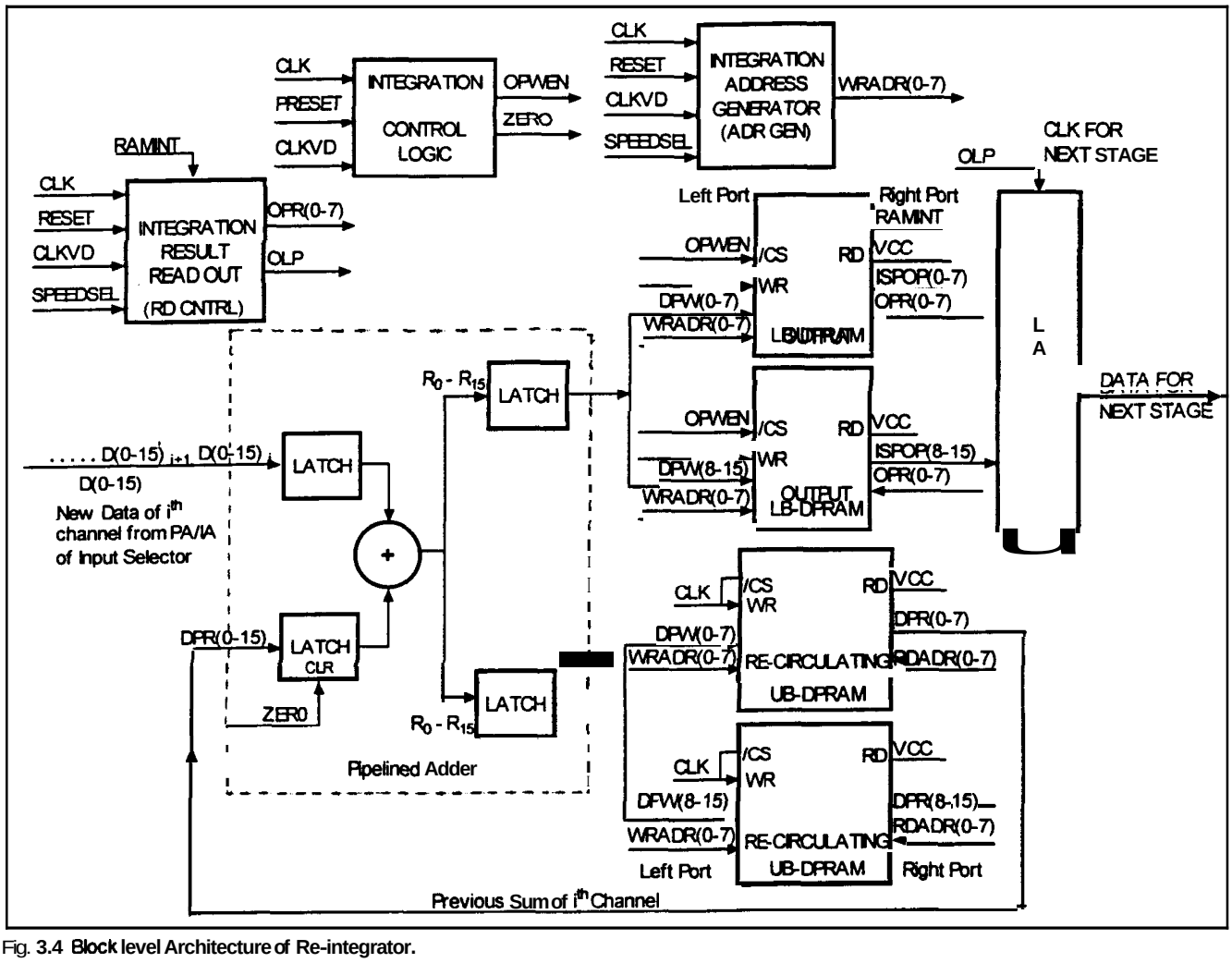


Fig. 3.4 Block level Architecture of Re-integrator.

zero. During the first cycle of every integration, a control signal makes the DPRAM terms at the adder inputs appear as zeros, so that the incoming number is simply reproduced at the output of the adder. The write control logic provides the address for the "write" side of the recirculating DPRAM along with the write-enable signal. The data from the adder gets written into the DPRAM. The pipeline delays in the adder are taken into account such that the address generated by the write control logic are synchronized with the time-multiplexed channel numbers. For the first set of **256** data points (corresponding to one spectrum), the read address remains at zero address, while the write address keeps incrementing, and the zeroing signal remains active, pushing the data as it is, to the DPRAMS. After the first frame of **256** channels, the read address counter starts incrementing along with the write pointer, and the zeroing signal is laid passive, so that the data of previous time frame are added to data of respective channels from the next time frame. These second iteration results are written in the address range **257** to **512** of the DPRAMS. Similar iterations ensue with a constant lag of **256** locations between read and write pointer of the DPRAM and the pointers round back to zero every **1024** counts so that the DPRAMS act as recirculators. Even though **512** locations suffice for this scheme, DPRAMS have been chosen to be 1K in size due to standard sizes available and to provide for future enhancement in number of channels. The total width of the DPRAMS is chosen to be **16** bits owing to the fact that with full integration (**256** frames) the result may grow by a maximum of 8 bits. The DPRAMs have an access time of about **50** nsecs, sufficient to complete reads or writes at the data rate of

16Msamples/sec. After the an integration of one data block is over, the results have to be sent out to the next section for further processing. With the minimum integration of 16 samples, a set of 256 numbers appear in a burst of 64 nsecs interval for a length of 16 msecs, once every 256 msecs. In transferring these results. two distinct possibilities aspects are worth considering:

a. The next stage is directly connected to the adder outputs and receives data during the last summing cycle of each pre-integration. In such a case, the entire logic that follows will also have to work in bursts at the same speed as the pre-integration section i.e., 16 MHz, even though the data rate after pre-integration has reduced the data rate substantially. This is inconvenient since it does not allow us to make use of the reduced speed which would in turn simplify the implementation of the later and save on device costs.

b. The data need not be sent directly to the next section, and can be it can be temporarily stored at a high rate using conventional memory (using high speed SRAMS) when an integration is over,. The data stored is later sent out slowly to the next sections, at a rate only fast enough to ensure that the transfer is over before the next set of integrated outputs become available. This may require an additional storage cycle at the end of each integration and require additional logic to control the read and write generations of the SRAM.

Considering these, as an alternative solution, a second pair of DPRAMS is connected in parallel with the first set, on the writing side. The write enable for these DPRAMS are generated only during the last cycle of the integration, with proper compensation of the adder pipeline delays, so that the final sums coming out of the summer get duplicated into these DPRAMS at full speed in the last integration cycle. After the last cycle, the control logic disables the write side of the "output buffer DPRAMS" and initiates a fresh cycle of integration by activating the zeroing signal again for one time frame. Thus, the output buffer DPRAMS get a burst of write pulses only during the last cycle of every integration. A separate logic generates an address sequence for the read side of the output DPRAMS, and reads out the data along with suitable clock pulses at a slower, "uniform" speed to the sections thereafter. The read out starts after one integrated spectrum is completely loaded into the DPRAMS and finishes the transfer before the completion of next integration. The rate at which the outputs from these DPRAMS are read out and pumped into the next section depends on the pre-integration, and is once every 960 nsecs at the minimum pre-integration and once every 16 μsecs at the longest pre-integration. The entire adder circuitry is hosted inside a single EPM 5128 EPLD chip, using about 80 gates and 89 flip-flops,, with two parallel buffered busses brought out of the chip to connect to the integration and output DPRAMs respectively, so as to simplify the PCB design and reduce the current drive per chip output. The address generation logic for the two sides of the integration DPRAMS is handled by another EPLD and the address management and clock generation for the read side of the output buffer DPRAMS and further stages is handled by a different EPLD chip. A third EPLD generates all the read-write control and synchronizing signals on the board. These three EPLDs are EPM5032, (15ns) and were chosen for above functions in order to accommodate maximum number of pins for the required density and speed for a given cost, with minimum PCB routing complexity. All control sequences are generated with counter based state-sequencers designed to work with a clock period of 64nsecs. The pre-integration counter is programmed

at setup time before an observation and presets automatically, every time the pre-integration is over. This control logic starts the address sequencing for the integrating DPRAMs after accounting for pipeline delay and also initiates a transfer sequence at the output DPRAM once every time the pre-integration counter gives an overflow. The transfer of outputs are stopped whenever an interrupt is received from the output DPRAM, indicating that the top most (also the last) location is read out.

Considering physical constraints in PCB layout for the circuits and limitations of high speed designs, it is preferred that the entire circuit of the ISSP, described above, is explained hitherto be located within a single board and the component spacing be minimized to avoid transmission problems. The search pre-processor logic that follows this module is to run slower by a factor of atleast 16 times and can be located in a separate PCB.

3.2.3. Search Pre-processor (SP):

This module receives data after pre-integration and performs all other functions required before recording the data. As shown in figure (3.5), the data sent from the DPRAMS of ISSP are passed through another sliding window which has an input width of 16 bits and output width of 8 bits, the top 8 most

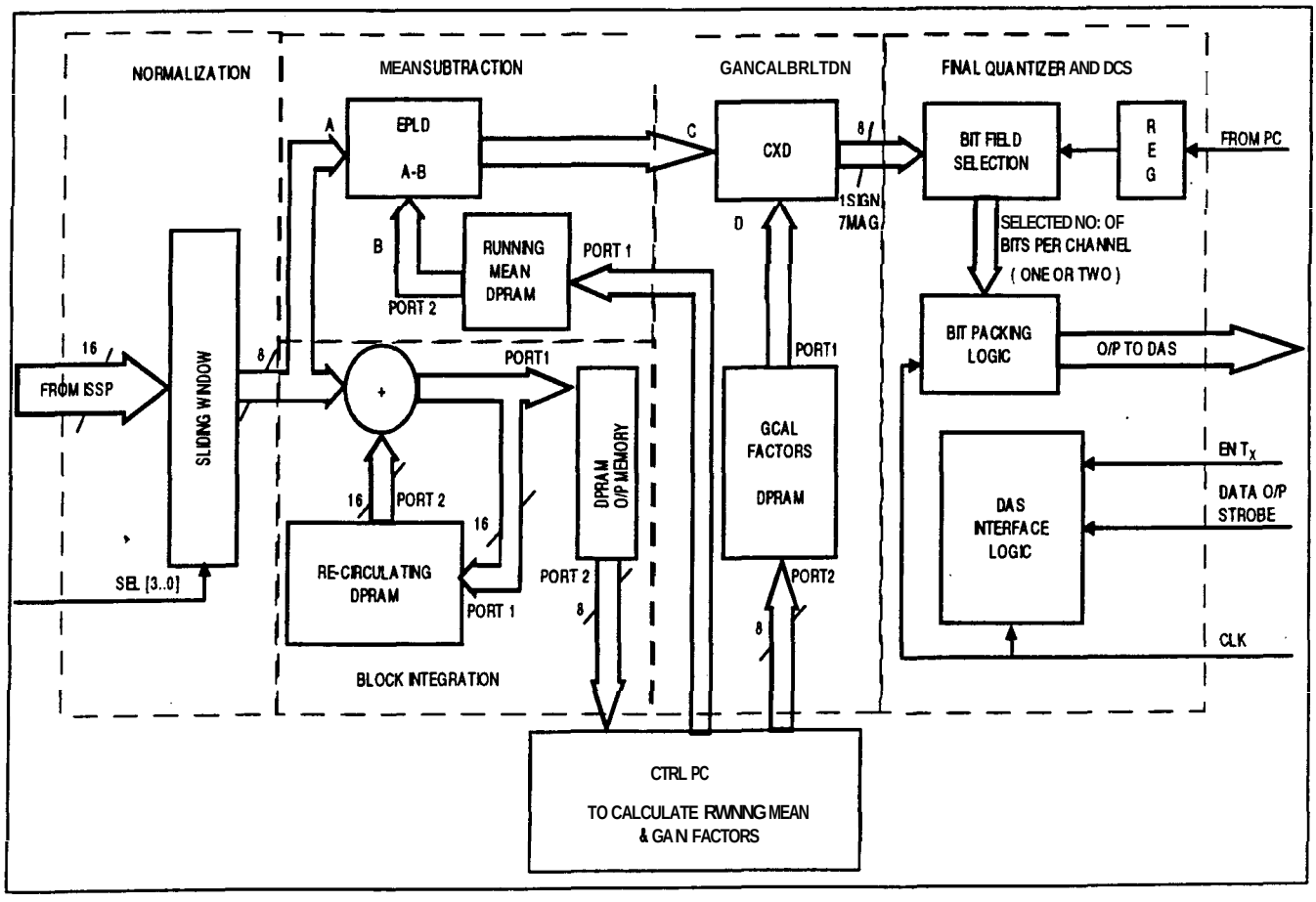


Fig. 3.5 Block level Architecture of Search Pre-Processor module.

significant bits, that are likely to be filled with nonzero data in the pre-integrated output are selected, using a multiplexer scheme similar to the one discussed in ISSP. The sliding window position is frozen for a given signal level optimally at the beginning of an observation. The sliding window allows any of the 8 consecutive bits to be selected under software control, among the sixteen bit pre-integrated data bus. The optimum selection depends upon the bit-growth related to the number of samples added together during pre-integration. A total of 9 possible window selections are encoded as a window number when selected. The multiplexer logic decodes this number for each data bit and selects the appropriate bit field. In order to avoid truncation errors, the bit field selected is rounded off with the most significant bit of the unselected field. The window selection can be programmed just before an observation. This stage serves as a gain controller for the signal band as a whole. The relative gains of the channels are accounted for by separate gain calibration logic (explained later).

After the sliding window, the data samples bits are connected in parallel to the block integrator and mean subtractor sections. The block integrator logic is identical to the architecture of the pre-integrator, but is used to continue the integration further after pre-integration, with a programmable length of 16 to 256 samples pre-integrated samples. This block integrator is also equipped with a pair of recirculating DPRAMS and a pair of output DPRAMS, which are it used in a similar way as in the pre-integrator and dumps the results in the output DPRAMS after every block integration. The address of the DPRAMs are chosen such that when the block integrated data of the last channel is written into the output DPRAM , a flag is generated from the DPRAM indicating the availability of the latest block integrated data. This flag is polled by a control PC/AT through a parallel port on its ISA BUS interface. A run-time control routine on the PC/AT then reads out the data from the DPRAMS and fills these values into a local array. Using these values, the PC calculates the running mean and gain calibration scale factors as follows:

3.2.3.1. Running Mean Calculation:

For a given frequency channel F, the running mean M, at a time when the N th block-integrated sample arrives, is given by:

$$M_{(N,F)} = \frac{1}{L} \sum_{i=(N-L)}^{N} d_{(i,F)} \qquad (3.1)$$

where L is the smoothing length in units of the block integrations, $d_{(I,F)}$ is the ith block integrated data for spectral channel F. It is easy to show that

$$L \cdot M_{(K,F)} = L \cdot M_{(K-1,F)} + d_{(K,F)} - d_{(K-L,F)} \qquad (3.2)$$

This form of the equation indicates that at any given time, it is necessary to store L consecutive spectra in a two dimensional array of the PC memory, but the computation involves only two operations:

summing the newly arrive data to the previous sum and subtracting the oldest element in the array from this sum. After this operation the new data is overwritten in place of the oldest element. This index corresponding to the new sample rotates over the length (L) of the array., which defines the time constant of the running mean. In this calculation, all the data points contributing to the running mean are added with equal weightage, leading to a rectangular window function. The low pass filter output is subtracted from consecutive pre-integrated samples 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. As mentioned in section (2.1.1.3.), it is sufficient to update the running mean at every block-integration interval. This reduces the running-meancalculation rate, providing an update at intervals of 128msec at the slowest rate. This is equivalent to sampling the low pass filter pass band at ≈120 times the Nyquist rate. Thus the leakage from the aliased components would be negligible. Using the above procedure, a new set of running means for all channels is calculated every time a new block-integrated spectrum is read into the PC.

3.2.3.2. Calculation of Gain Calibration Factors:

The gain (band-shape) across the of the spectrum, is in general not uniform and should be equalized before the data in all channels can be summed up during post-processing. To calculate the scale factors by which the samples of individual channels have to be multiplied to equalize the gains, the running means evaluated above are used. The gain scale factor for each channel is computed as follows:

$$S_{(K,F)} = \left[\frac{\max(M_K)}{M_{(K,F)}} \right] \qquad (3.3)$$

where max(Mk) is the running mean of the frequency channel that has maximum gain. These scale factors are used to scale the data on each channel. Since the mean subtracted data occupies a maximum range of 24dB (corresponding to 8-bit representation), it is sufficient to have 8 bit scale factors. The gain scale factors are rounded to 8 bits and fed back to the SP module for scaling operation.

3.2.3.3. Feedback System:

These mean values and gain scale factors have to be fed back into the SP without disturbing its operation's. This is achieved by using another set of DPRAMS on the SP module. One side of the DPRAMS is connected to a read out logic which presents data to the mean subtractor and gain calibration block, while the other side is accessible from the PC to load new parameters once in a while. One DPRAM is to contain the running mean of all the channels and another stores the gain calibration factors. Each DPRAM has two banks of 256 locations, at any given time one bank will be available for the PC to dump new values, while the other bank is being read out at full speed by the SP board. When the PC completes dumping the new set of values, it sets a flag indicating to the SP module that new factors are ready for use. This bit also indicates the bank number in which the new factors are stored. The PC toggles the flag bit every time it loads a new set of values. A counter based control logic on the SP board polls this status flag at the end of every spectrum and

latches the value of the flag. The true and complement values of this latched bit form the most significant address bits of the two ports of the DPRAM, as shown in figure (3.6). Thus a change in the flag bit automatically switches the banks between the PC and the mean subtractor block. The read out logic synchronizes the read address such that at any given time, the scale factor/running mean and the incoming data sample correspond to the same frequency channel. This will happen in complete synchronism with the system clock and not cause any delay in the flow of input data and scale factors to the multiplier. The PC gets a flag from the controller indicating that the banks have been switched, so that the PC can now use the first half of the table to update the next set of values. This switching of the address map is transparent to the PC side and it appears as though only a single bank of 256 locations available for programming at any given time.

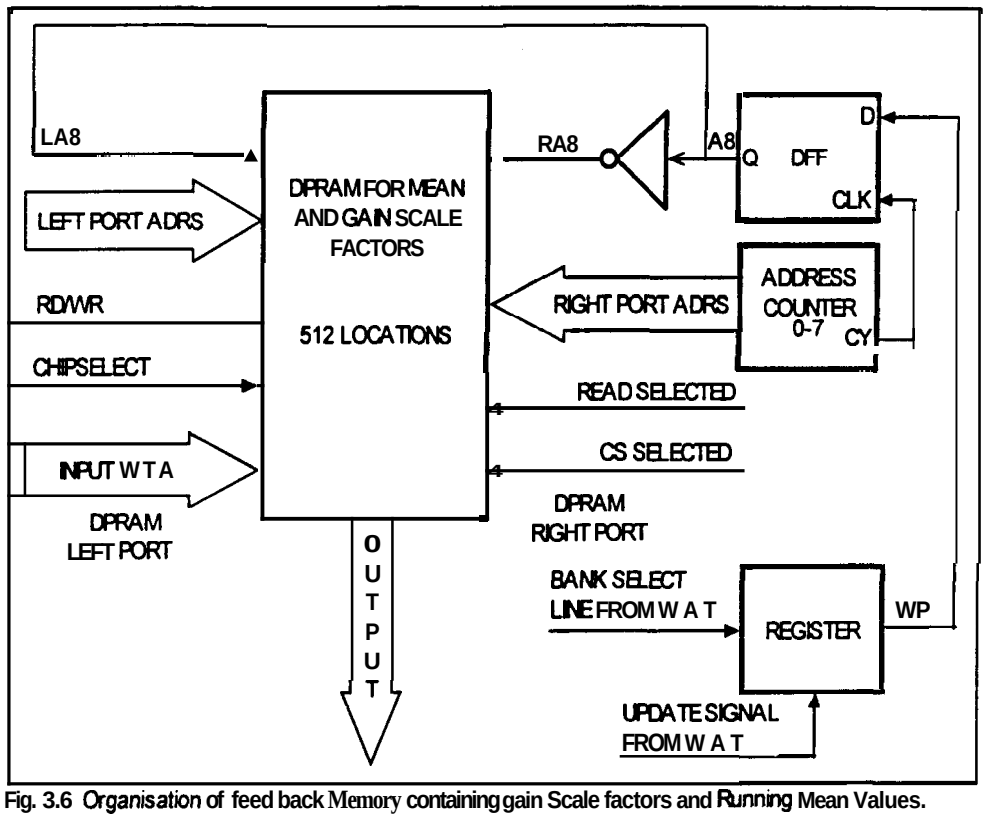


Fig. 3.6 Organisation of feed back Memory containing gain Scale factors and Running Mean Values.

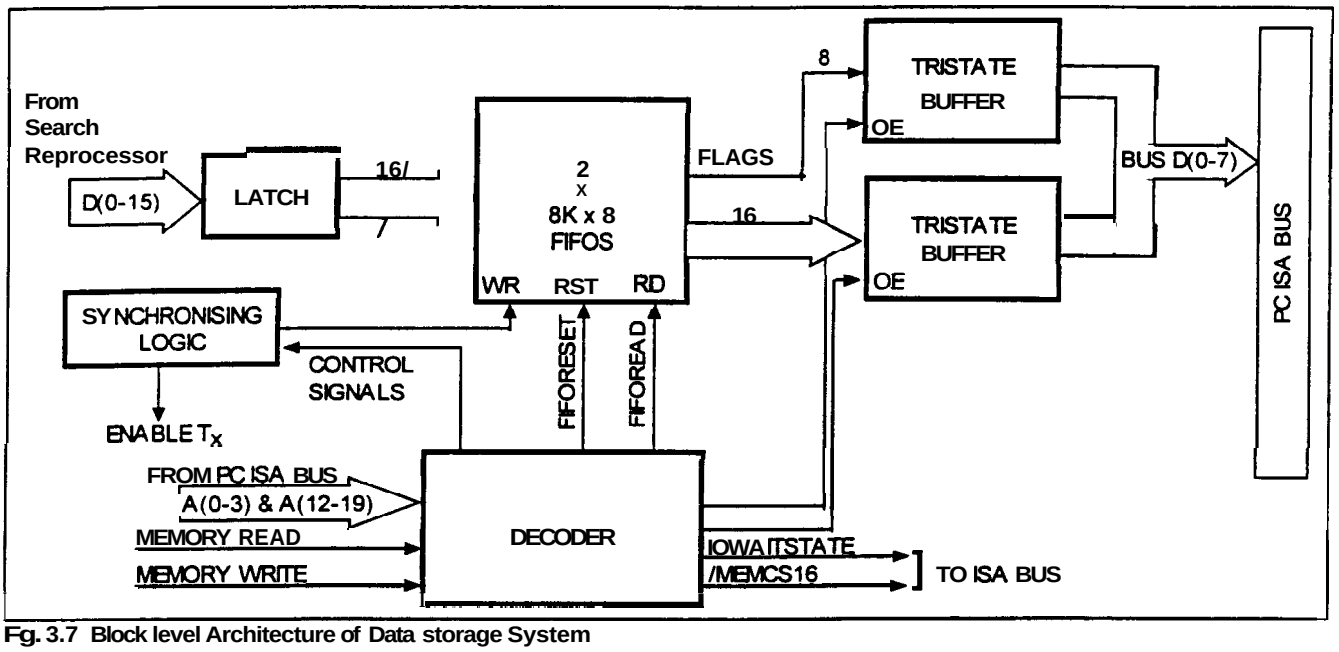
To subtract the means of consecutive channels from their respective pre-integrated data samples, a pipelined subtractor is used. This logic calculates the difference of two 8-bit inputs (data-mean) and rounds off the difference to 8 bits. The outputs of the mean subtractor are then passed on to a GAIN calibration EPROM look-up table. This EPROM is programmed such that the mean subtracted data forms the lower address field and the gain scale factor forms the higher address field. The locations addressed by the combinations of these fields are pre-programmed to store the corresponding products of data and scale factors. The pre-calculation of the contents of the EPROM also rounds the 16 bit product to 8 bits, which is sufficient to retain the dynamic range of the mean subtracted data. After gain calibration, the data is latched onto a bit-field selection and packing circuit. The initial sliding window logic before block integration and the pipelined mean subtraction logic are housed in a single EPM 5128, 25nsec EPLD consuming about 162 gates & 59 flip-flops.

3.2.3.4. Bit Field Selection Logic (BFSL):

As mentioned in sec. (2.1.1.3), the data at this stage (after mean subtraction) may be quantized to just one or two bits, retaining since more than 80 % of the original signal-to-noise ratio information in the power spectra. The saved bit-width can be bargained for extra bandwidth or time resolution in future upgradations. The bit - field selection logic is programmed to select the sign bit together with any one of the magnitude bits. This choice is usually based on the rms deviations in data which can be measured in a test observation before the actual observation. The selection logic is similar to the sliding window explained earlier, except that the sign bit is always chosen and the sliding window is only one bit wide to select any bit, within the magnitude field depending on the strength of the signal after gain calibration. The bit field selection & packing logic is implemented along the address generation logic for block integrator in a single EPM 5128 (25ns) EPLD utilizing about 69 gates & 64 flip-flops.. For two-bit mode, the design of bit-field selection is modified to choose 2-bit frames and the chip has to be replaced (the two designs are pin-compatible). After this stage, the data can be recorded on to a PC- based data storage system (DSS) . The DSS is interfaced to the PC on ISA bus. Since the ISA bus offers a 16-bit data bus, it is preferrred to pack the selected 1-bit or 2-bit fields into 16 bit words and store them on the DSS. This results in enhancement of effective data acquisition rate and minimization of data storage space, both of which are important. A counter based timing logic latches the bit-fields into consecutive pairs of flip-flops of a shift register and loads the entire 16- bit word onto an output register once the word is filled. The bit-field packing logic generates a write pulse to the DSS every time a new 16- bit word is read at its output. However, the transmission of the data and write pulses starts only when the DSS sends a flag, indicating that it is ready for recording. The DSS can go through its setup and diagnostics initially and then enable the flag. The BFSL control logic synchronizes the data transmission such that the transmission starts only with the first channel of a new spectrum, after enable flag is set by the DSS.

3.2.4. Data *Storage* System (DSS):

For reasons of simplicity and availability of a large base of software and hardware in the market, a PC/AT based DSS was built to acquire and record the data from the SP module. The DSS is equipped with a PC/AT mgther board, 2x1.2 GB hard disk drives and an EXABYTE video tape recorder, apart from conventional interfaces such as a keyboard, monitor, etc.. To acquire the data and send it to the hard disk, a separate card is built with the architecture shown in figure (3.7). This card is mapped onto the memory of the PC between the address range 0xd0000 to 0xe0000, which is usually the space allocated for user interface development on the standard PC architecture. Memory mapping allows block transfers to the hard disk using standard device drivers of the operating system and enhances the speed performance. The PC can configure the DSS initially to a default control state and then enable the transmission by setting the enable flag on the DSS to inform the BFSL that it is ready for receiving data. The data and clock arrive in differential transmission links from the SP board, owing to the fact that the DSS and SP may not be physically nearby and the path may be prone to local interference. FIFOs are used to temporarily store the arriving data. The



Fg. 3.7 Block level Architecture of Data storage System

data flows from the instrument into the data acquisition card in bursts with each burst containing 256 words (16 bit) along with the associated write pulses. The average data rate at the minimum pre-integration with 2-bit quantization is about 256 Kbytes/sec. Typically the hard disk seek & latency time is of the order of 20ms and each track of the disk may consist of about 64bytes of storage capacity. Typical observation sessions may run for about an hour continuously, amounting to about a GB of recorded data. Once the head is positioned, the data can be stored into the disk at a much higher speed. The head has to move to the next track every time, 64Kbytes of data'is stored, thereby drastically reducing the average throughput. It is obvious that the speed may be sustained at a higher rate by buffering the data using the FIFOs during the seek & latency intervals. At the time of design, hard disks with a capacity of 1GB and above were available only with SCSI (Small Computer Systems Interface) standard. With this standard the fastest transfer speed sustained by a hard disk was about 512Kbytes/sec.The clock writes the data into a pair of FIFO memory chips. The FIFOs generate empty, half-full and full flags corresponding to the number of locations filled in with the incoming data. The PC periodically polls these flags and whenever it senses a Half-full flag, it initiates a transfer to the hard disk. The program instructs the disk controller to record, from the starting address of the FIFOs, the data of block size equal to half the FIFO size. The Hard disk controller then takes over the ISA bus and generates address in the specified region and acquires the data and transfers it into the hard disk. An advantage of using FIFOs is that FIFOs of different densities fit into the same pin outs, as long as the data word width remains the same. Thus, the FIFO size can be scaled up from 256 locations to 32K locations available presently based on the seek and latency of the hard drives without any change in the interface.With laboratory experiments, it is found that for the normal data recording rates expected from this instrument (about 128 Kbytes per second) can be sustained while acquiring onto the disk with a pair of 8K location FIFOs.At this data rate, half the FIFO gets filled in 64ms and the PC stores this data into the disk in an interval of 32 ms on the average. Thus, the PC gets about 50 percent free time to handle any other job.The entire control logic of the DSS is fit into a single EPM 5128,(25ns) EPLD consuming about 69 gates and 24 flip flops.

3.3. System Software:

All the routines for diagnostics, setup and on-line computation are written in C language for obvious reasons of simplicity and closeness to hardware level operations, which are essential for a real- time control system. DOS was chosen, in preference to many other real -time operating systems, as the base operating system owing to the large base of user interface application software and drivers for different hardware interfaces that may be used on the PC/AT platform. The Control PC and the DAS PC are linked together via serial port interface, and interrupt driven routines are written to receive and transmit data via these ports. The flowchart of figure (3.8) shows the sequence of operations for an observation session. During the beginning of an observation, the routine procedure on the control PC includes can performing all the diagnostics and setup operations for the observation, waiting for the start time (given by the PC's time and date) and then sending a message via the serial port to the DAS to start acquiring. The DAS then activates its "enable transmit" line to the SP board, and returns a message to the control PC that it is ready. The control PC then releases the clock to the system by enabling a gate that controls the clock distribution to the entire system. The data then flows from the sampler to the FFT, ISSP, SP and DAS. The control PC, besides doing the on-line computation jobs (namely, of calculating the running mean and the gain calibration factors and feeding them back to the SP board), also polls the system time and date repeatedly to check if end time is reached. If so, the control PC disables the clock and informs the DAS PC program via the com port to stop acquisition. The DAS program can then close the result file and exit. The data acquisition can proceed un-interrrupted, limited only by the capacity of hard disk, which in the present case is about 2 GBbytes, corresponding typically to about four hours of observing span. After the disk is full,

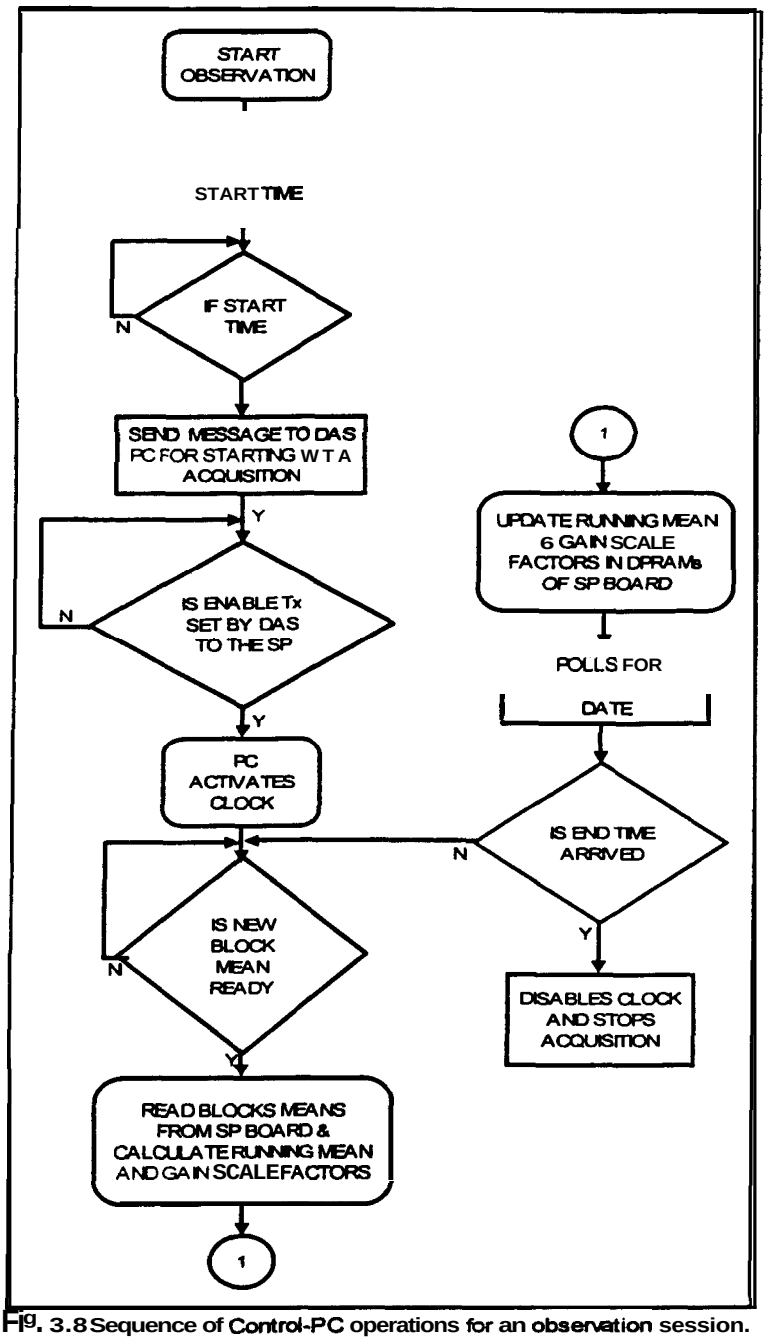


Fig. 3.8 Sequence of Control-PC operations for an observation session.

the observation will have to be stopped and the data on the hard disk may be backed up to a video cartridge using the EXABYTE tape drive and the tape can be transported to a remote post-processing site for further use. Also, for preliminary test observations and calibrations, it is required to have a local facility at the telescope site itself for conducting basic post processing on the observed data. An Ethernet link is installed on the DAS unit using which the data on the tape can also be ported through the standard file transfer protocols (FTP) onto a local workstation, where the post processing tests can be continued, while the data of the next observation are being acquired.

3.4. Test and Results

Simulation tests have been were done in the lab, in order to check the behavior of the circuits. Once the simulation tests were successful, the system was fabricated and checked by feeding digital pattern test in the laboratory. After these tests showed that the actual system compiles with the simulations, the system was moved to the Ooty Radio Telescope and connected to the telescope. Test observations were then conducted on actual pulsars.

The ORT was equipped with the analog receiver, sampler and FFT system that could handle a bandwidth of 8MHz, as explained in chapter 1. The array combiner available at this time was for combining outputs of 4 antennas. Even though the ORT provides only one output from the entire array using analog power combiners, the AC was used to provide the required format conversion of data between the FFT module & the PSP. All input channels of the AC outputs were fed with common inputs since the ORT provides only one polarization. After the PSP was interfaced to the ORT base-band receiver and FFT system, the machine was run with different pre-integration and block integration values. The running mean calculated was logged into files and examined for obtaining suitable sliding window positions matching the base-band power levels at ORT. Figure (3.9) shows the band-shape, obtained after optimizing the sliding window positions. The band shows a ripple of about 5dB, and matches the measurements on a spectrum analyzer.

A test setup was used to simulate pulsar like signals locally to check the stability of the entire chain (figure(3.10)). The test setup has two channels, one of which is connected to the Antenna's IF output while the other gets its input from a broad-band noise generator. A band-pass filter reduces the pass band to 8 MHz around the center of frequency 30MHz. These two signals are then fed to the IF sections of the ORT for converting the IF to a base-band output. The local oscillator going to the noise generator channel is amplitude modulated with a pulsed signal from a function generator. The modulated base-band noise in this channel was added to that coming from the antenna and fed to the sampler after sufficient video amplification to match the dynamic range of the A/D converter. With this setup, the entire machine was run with the antenna tracking fixed point in the sky. The recorded data was reduced during off-line processing by dedispersing and collapsing all frequency channels for a DM of 0. The recorded data of all channels was added and folded in time over an interval equal to a multiple of the modulation period, so as to see if pulses get folded synchronously and ensure that no spurious interference would be falsely detected as a pulsar signal.

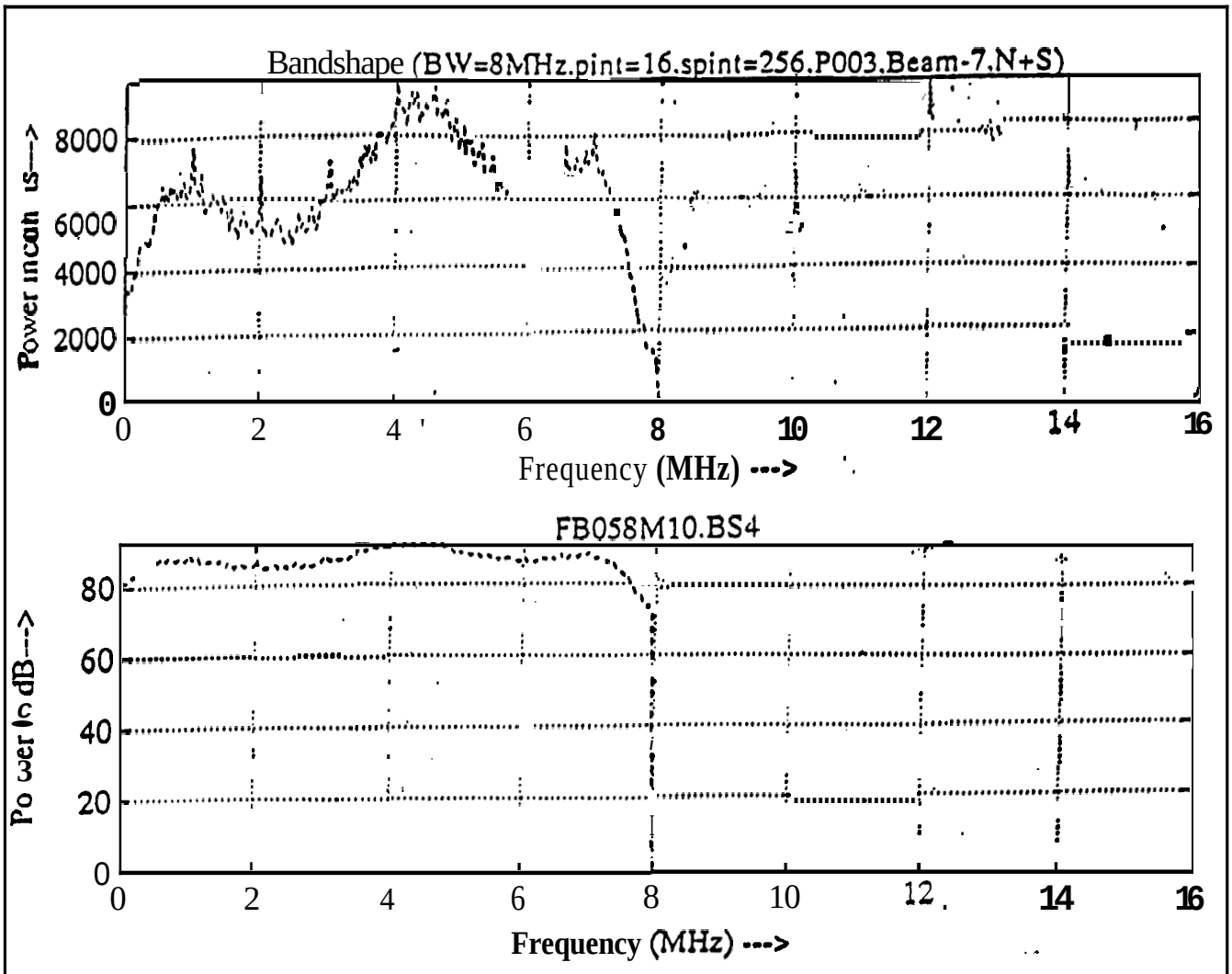


Fig. 3.9 Block - Average power spectrum obtained by Pulsar search preprocessor using ORT.

Observed Bandwidth : **8 MHz**
Number of Spectra averaged : **4096**

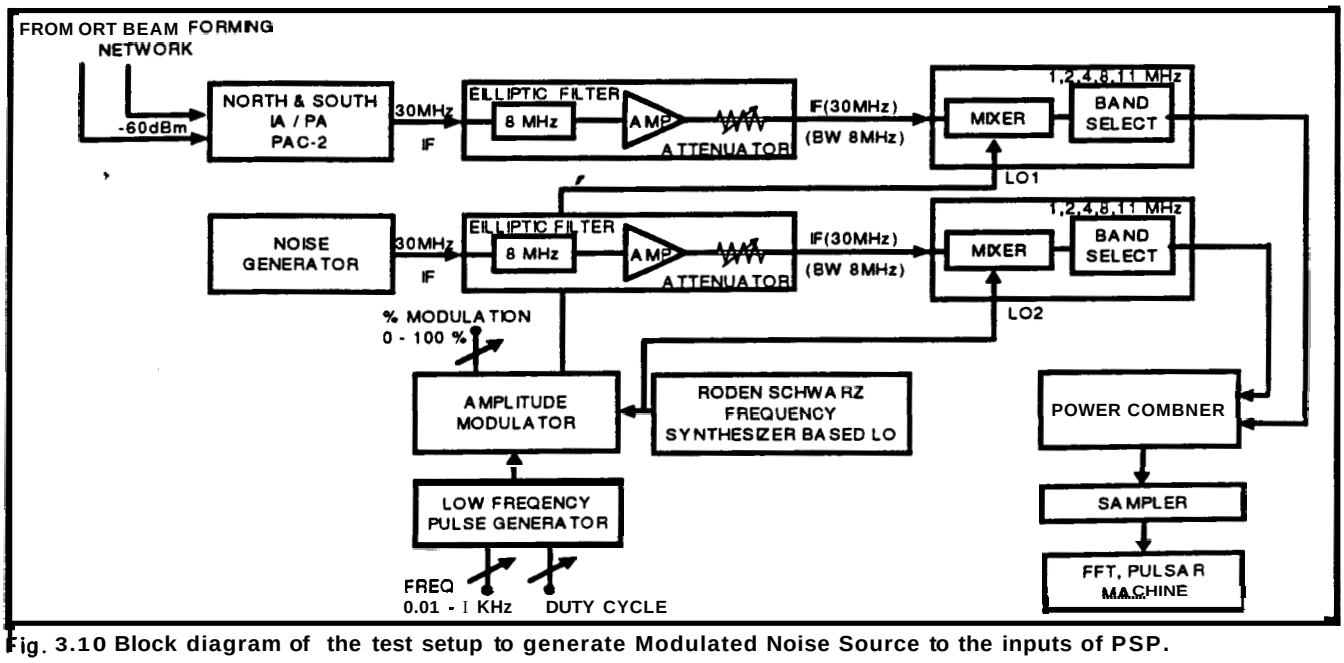


Fig. 3.10 Block diagram of the test setup to generate Modulated Noise Source to the inputs of PSP.

Figure (3.11) shows the folded profile of one such run. The modulation period was known approximately and hence some Smearing is also seen in the folded profile . After this, a trial observation was conducted on a

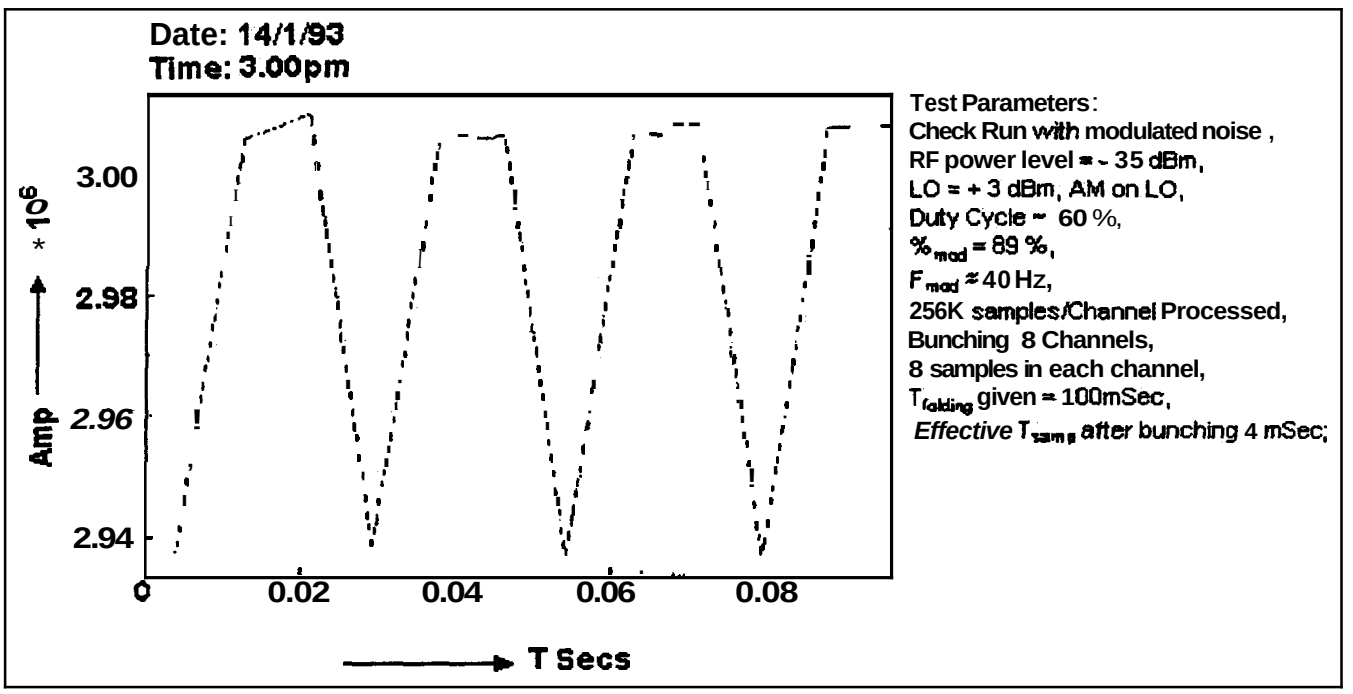


Fig. 3.11 Folded profile from modulated noise test.

reasonably strong pulsar (PSR0740-28). The channels were dedispersed and added together and was inspected for pulses with deflection beyond about 7σ level of the fluctuations. These were used as valid pulses and the observed time gaps between pulses were compared to those expected. The residual error was then attributed to an error in the assumed clock period .The data was refolded using the new sampling interval value and substantial improvement in the pulse sharpness of the shape and the pulse strength were observed. The enhancement of signal to noise ratio after folding was found to be as expected. Figure (3.12b) shows the dedispersed, raw profile while figure (3.12a) shows the profile after folding about 12 periods.

Then several pulsars were observed to find the detection limit in terms of the minimum detectable flux density with this system at ORT. The data were also Fourier transformed over long sequences and the sources of local interference (which would have their best signal-to-noise ratio for de-dispersion with DM = 0) such as power-supply 50Hz leakages, influorescent lamps, display terminals, etc., were identified. The grounding/shielding of the PSP was improved to minimize the levels of these interferences. Figure (3.13) shows one such spectrum of a weak pulsar (20mJy flux density) where the leakage from power supply rectifier is identifiable at a level comparable to the strength of the pulsar signal. After a series of successful observations, it was attempted to calibrate the profile in flux units considering the architecture of the machine and statistics of the pulsar signal (development of the procedure used is illustrated in appendix(B). Presently, continuum calibration sources and the reference cold-sky regions have been chosen at the same declination

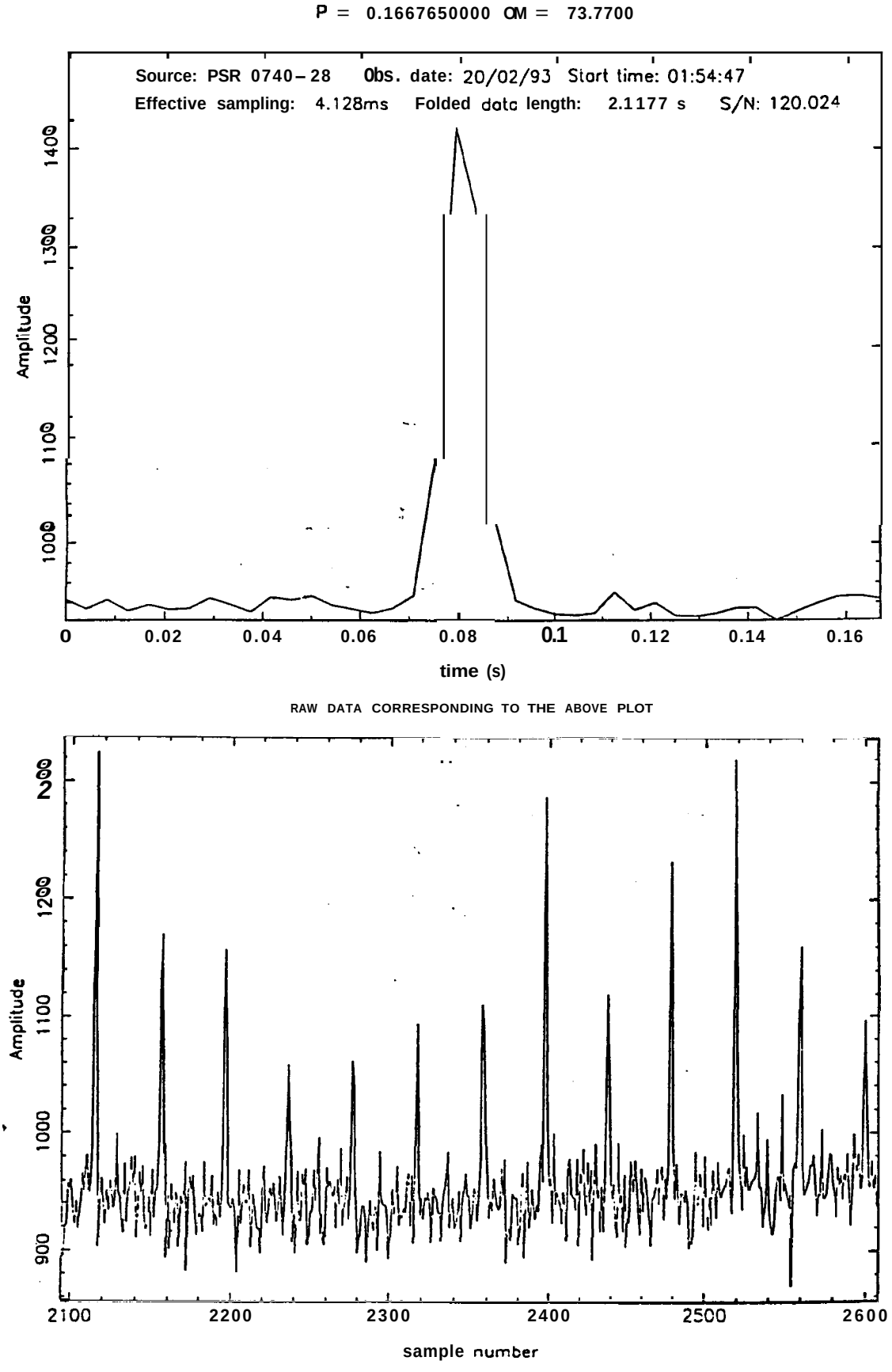


Fig. 3.12 Folded profile and Individual pulses (12 periods) of Pulsar 0740-28 Observed at ORT using PSP

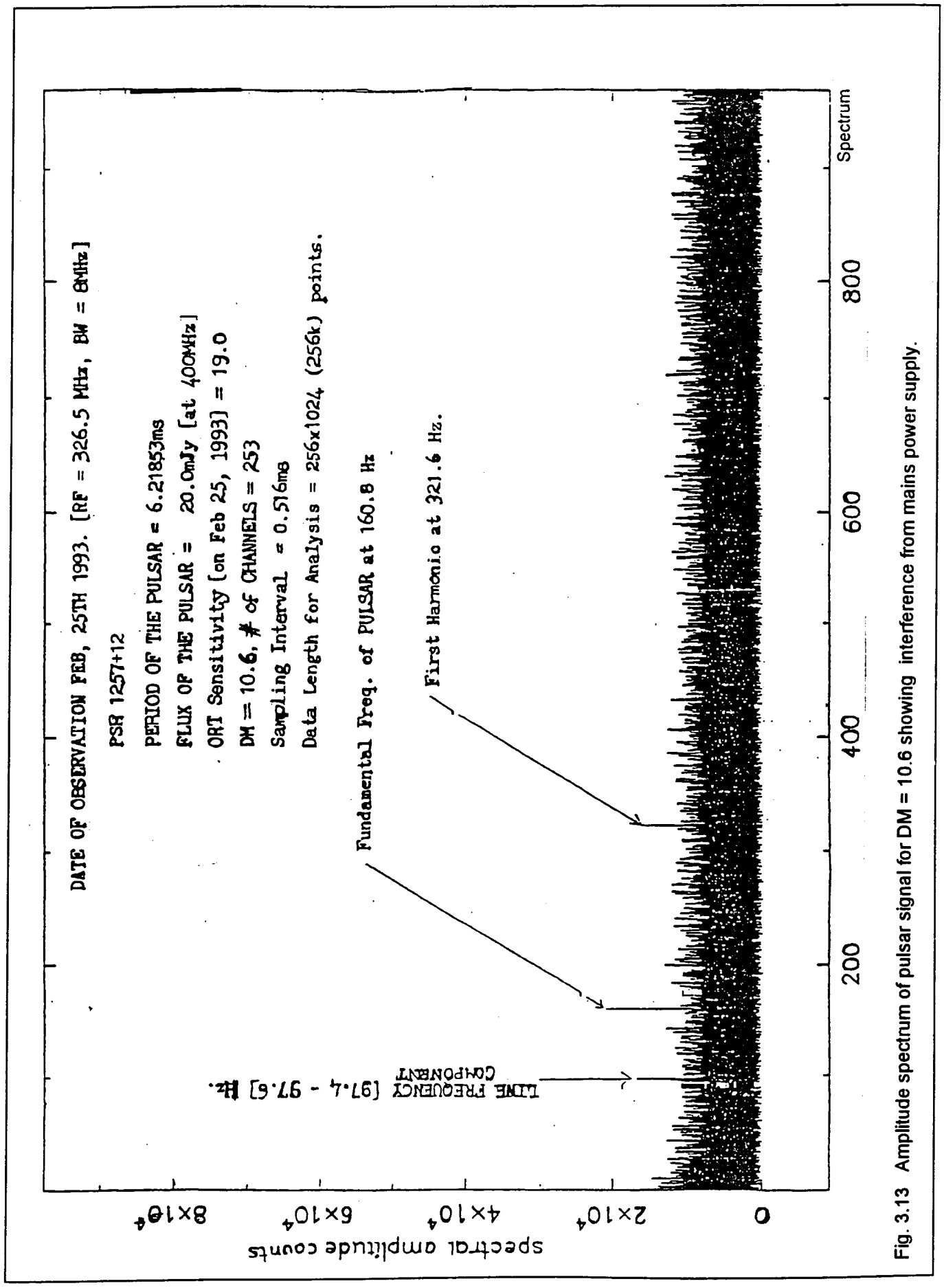


Fig. 3.13 Amplitude spectrum of pulsar signal for DM = 10.6 showing interference from mains power supply.

as those of the pulsars to avoid any declination dependent factor entering into the calculation of sensitivity for a calibration. Further, to avoid nonlinear quantization effect at FFT and power detector stages, sources with flux densities less than a few Jansky are chosen. These effects are considered small, and ignored presently, but need to be measured to allow for more accurate flux measurements. Results for some pulsars observed with a sampling interval of 0.5 msec (pre-integration of 16 samples), one-bit quantization mode are shown in figures (3.14a), (3.14b) and (3.14c). Vertical bars in the respective profiles indicate the rms error due to the noise in the profile.

Figure(3.14a) shows a profile of PSR 1237-41, chosen for its low flux density ($S_{avg} \approx 2.5mJy$). It has a DM of 44 $pc \cdot cm^{-3}$ and a period of about 512 ms. Figure(3.14b) shows a profile of PSR 0740-28, chosen as an object with high flux density ($S_{avg} \approx 290mJy$), DM of 73.6 $pc \cdot cm^{-3}$ (for which the dispersion smearing within the chosen channel bandwidth will be close to the sampling interval) and period of about 166 ms. Figure(3.14c) shows a profile of PSR 1257+12, pulsar with a short

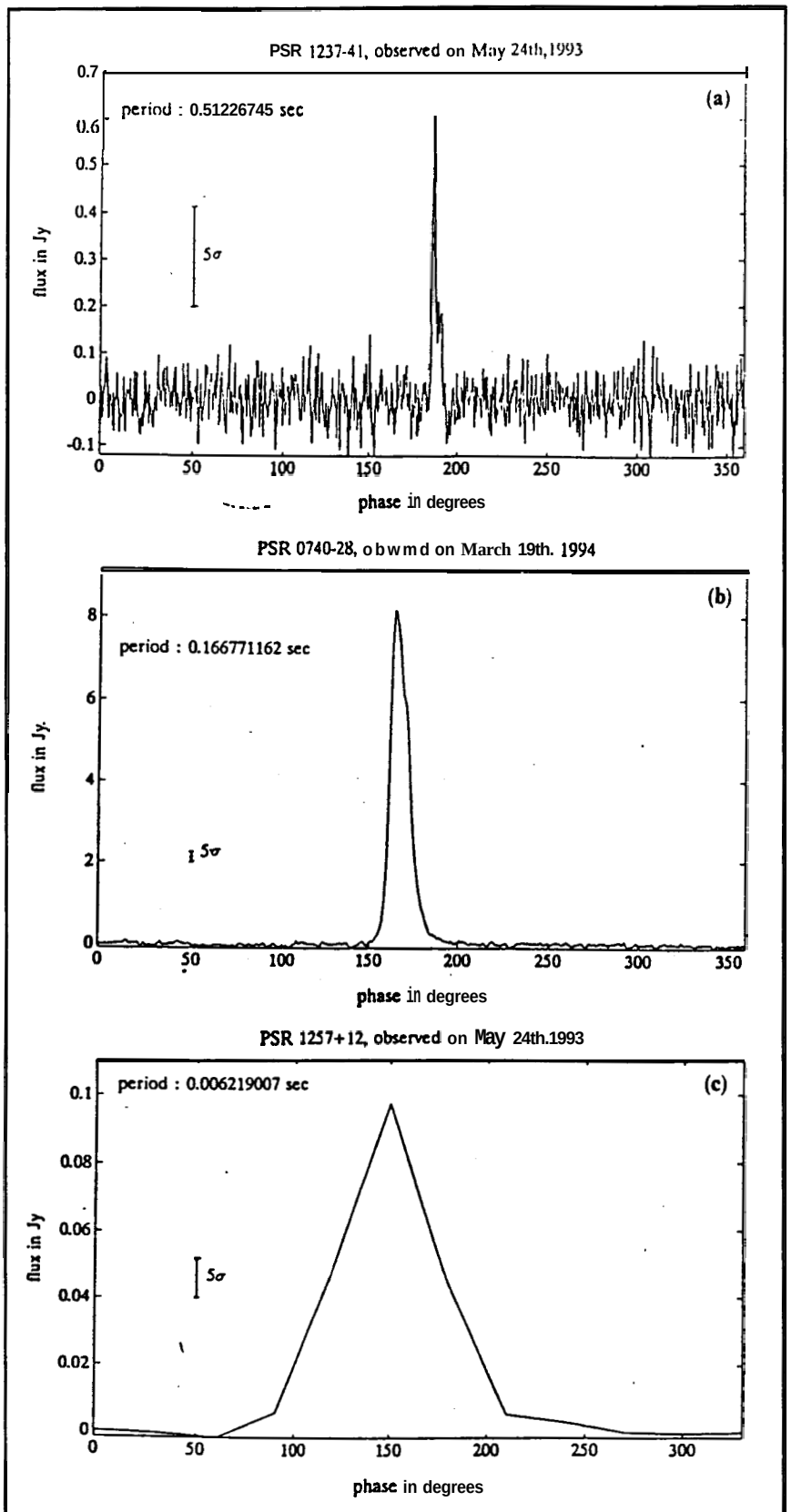


Fig. 3.14 Folded Profiles of pulsars observed at ORT using PSP (Published results)

period of about **6ms**, low flux density ($S_{avg} \approx 15mJy$) and a low DM of 10 $pc \cdot cm^{-3}$.

A paper was published describing this work and results (**Ramkumar.et.al**, 1994). This instrument has also been successfully used for pulsar timing observation over the last few years (**Indrani.et.al.**, 1997). Currently a survey for searching new pulsars is on at the ORT. Two more identical systems were fabricated and installed at GMRT successfully. The instrument is presently serving to help in testing various front-end modules at GMRT. Figure (3.15) shows a pulsar profile obtained at the GMRT with the array combiner adding

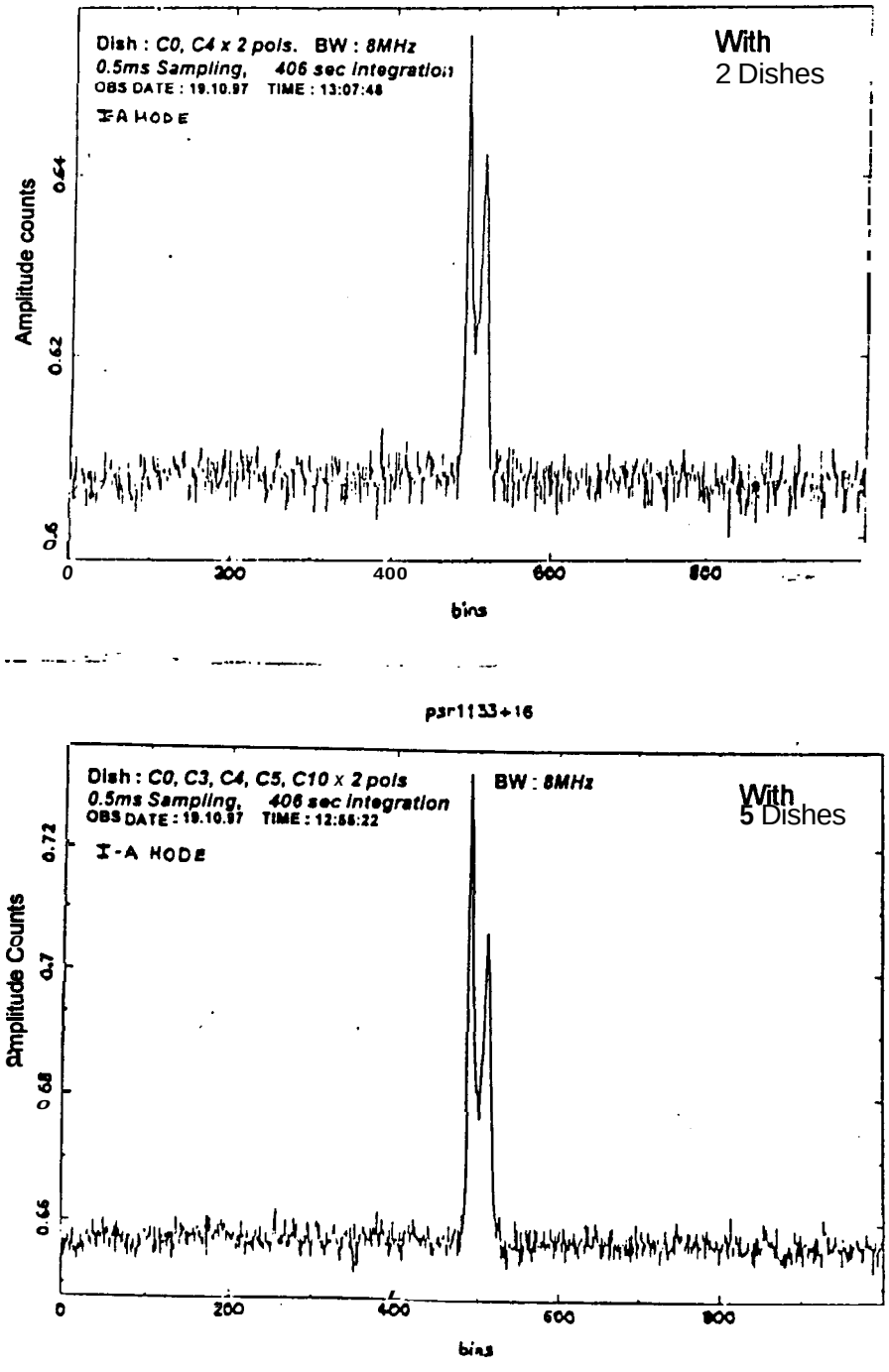


Fig. 3.15 Pulsar 1133+16 observed at GMRT using PSP

the outputs of multiple dishes in the IA mode. Once the GMRT becomes fully operational, the PSP can be used to conduct pulsar search with much higher sensitivity.

3.5. Portable Pulsar Receiver (PPR):

Some pulsar observations may require data with high time resolution limited only by the Nyquist sampling rate by the chosen band-width. Also, in some applications it may be preferred to have a band-width studied with very high frequency resolution to allow for finer dedispersion correction to study the details of micro-pulses and sub-pulses of strong pulsars. In such cases, the base-band voltage signals may be digitized to one or two-bits and recorded without any pre-processing using a high speed data recorder. The recorded data may be processed off-line to perform operations such as coherent dedispersion, pulse folding, etc., using suitable computers. To facilitate these types of observations modules developed for the pulsar search pre-processor were identified, configured suitably and used in designing a portable PC based data recording system.

The PPR is equipped with a two bit 4-level quantizer (as shown in figure (3.16)). To digitize the base-

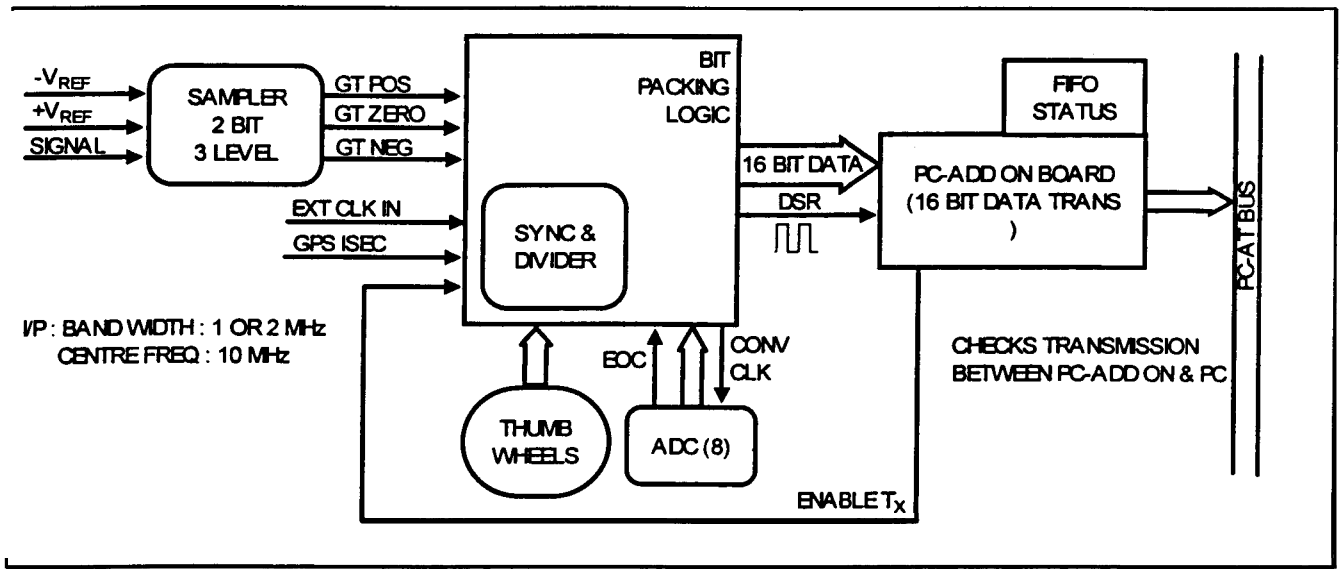


Fig. 3.16 Block Level Architecture of Portable Pulsar Receiver (PPR).

band signals, the quantizer has 3 comparators with thresh-holds set at $+V_t$, $-V_t$ and 0 V respectively (where V_t can be set in the range 2.5V to 0.2V). The comparator outputs then passed on to a encoder to assign digital codes corresponding to the regions depicted in table (3.3). The raw signal at the input of the comparator is in the form of zero-mean Gaussian noise voltage. The gains in the receiver stages that precede the quantizer are so adjusted as to fit the $\pm V_t$ thresh-holds at $\pm\sigma$ respectively, of the base-band noise.

Table (3.3)

INPUT			OUTPUT		
GT_NEG	GT_POS	GT_ZERO	+VE BIT	-VE BIT	STATUS
0	0	0	0	0	Vin < -Vref
1	0	0	0	1	-Vref < Vin < Zero
1	0	1	1	0	Zero < Vin < Vref
1	1	1	1	1	Vin > Vref

NOTE: Other combinations are not valid

A bit packing logic (shown in figure (3.17)) packs 8 consecutive 2-bit samples into 16-bit words to match the Bus-width of the Data Collection System (DCS). This reduces the output sample rate by the factor of 8. The counter based sequencer identifies 8 bit packing states to pack the samples and then latches the word to an output buffer. The sequencer also generates a write pulse for every output word that is transmitted along to the DCS. The separate counter produces a pulse for every 4096 output words. This pulse is used to trigger the set sequencer to generate an extra write pulse accompanying which a known 16-bit pattern is sent on the output data lines. This way, the recorder gets a marker word after every block of 4096 data words. This will be useful during post-processing to identify blocks of data with missing or extra data words. The 16-bit marker pattern has its top byte filled with zeros and the lower byte consisting of actually the output of another 8 bit counter that counts the number of such blocks of 4096 samples (module 25%). The bit packing and marker generation functions are designed into a single EPM7128 (15ns) EPLD. The outputs are driven to the DAS using TTL differential buffers. The clock for the bit packing logic is derived from a suitable oscillator through a programmable counter so that the sampling speed may be varied. Before starting to sample the data the timing logic waits for the DAS to give an "Enable Transmission" signals on the on-set of 10sec pulse from a global positioning system (GPS) interface (Chanthrasekaran, 1995). This synchronization is necessary in cases when simultaneous observations are to be conducted at different sites. The synchronization ensures that the start of .observation defined at the start time in the format of xx.yy.zz (hours.minutes.secs) where zz may be 00, 10, 20, 30, 40, 50.

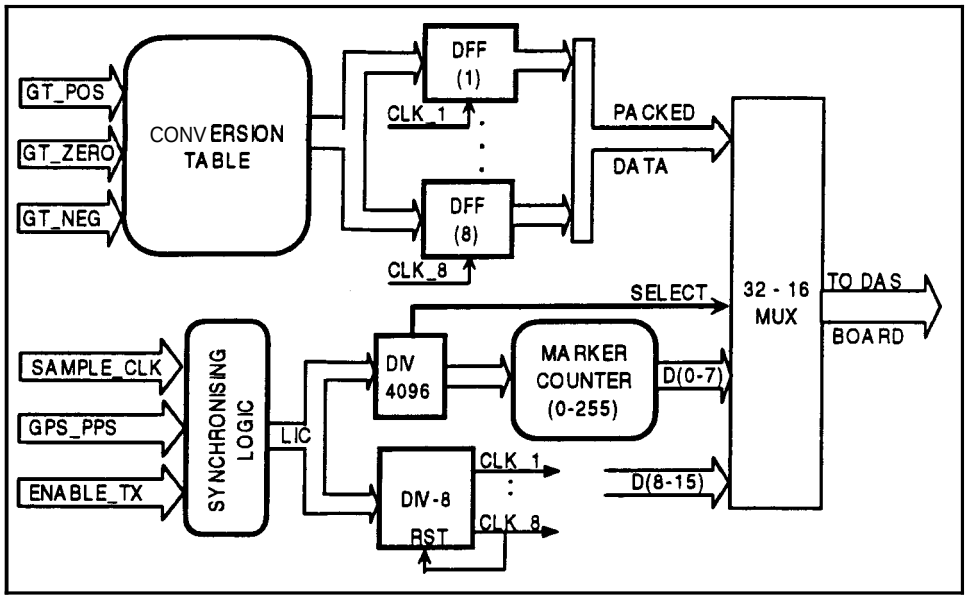


Fig. 3.17 Block Diagram of the BIT Packing Logic for PPR.

The PC time is synchronized to the observatory clock periodically. At the start of an observation the DAS locks the starting time and enables the bit packing logic through the "Enabled Transmission" line. At the on-set of the next 10second pulse from the GPS interface the data starts flowing into the DAS FIFOs and gets recorded on to the hard disk. The limiting factor for acquisition speed is set only by the hard disk access time. With this system, the maximum sustained rate of recording corresponds to about 0.75 Mbytes/sec corresponding to recording of 1.5 MHz band-width with 2-bit digitization at Nyquist rate.

Post-processing:

The recorded data is extracted and searched for the marker pattern in order to identify any slips that move its location during recording. The missing samples are inserted with equivalent samples of random noise (this does not affect the signal statistics significantly since the fraction of the data missed due to slips is expected to be less than 0.01%). This process can thus tag the obsolute time associated with the train of samples correctly.

The data is then Fourier transformed to produce narrow channels of optimized band-width based on the DM to be corrected. Incoherent de-dispersion is performed on the complex frequency spectra as explained in section (2.1.). The de-dispersed spectra are then collapsed to form a single channel and synchronously folded over the pulse period to achieve the required enhancement in signal-to-noise ratio.

This technique has been used and several observations have been conducted at ORT. Figure (3.18) shows the profiles of these observations. Figure(3.18a) shows a train of indiviudal pulses from pulsar 0950 ± 08 observed using the Ooty Radio Telescope with a base-band width of 1 MHz. Figure(3.18b) is a de-dispersed, time folded profile of pulsar 0950+08 observed with the same set-up. Several pulsars have been observed successfully using the portable pulsar receiver at Mauritius Radio Telescope (Issur,N.H., 1997). Currently a paper is being written for publication describing the systems and the test results.

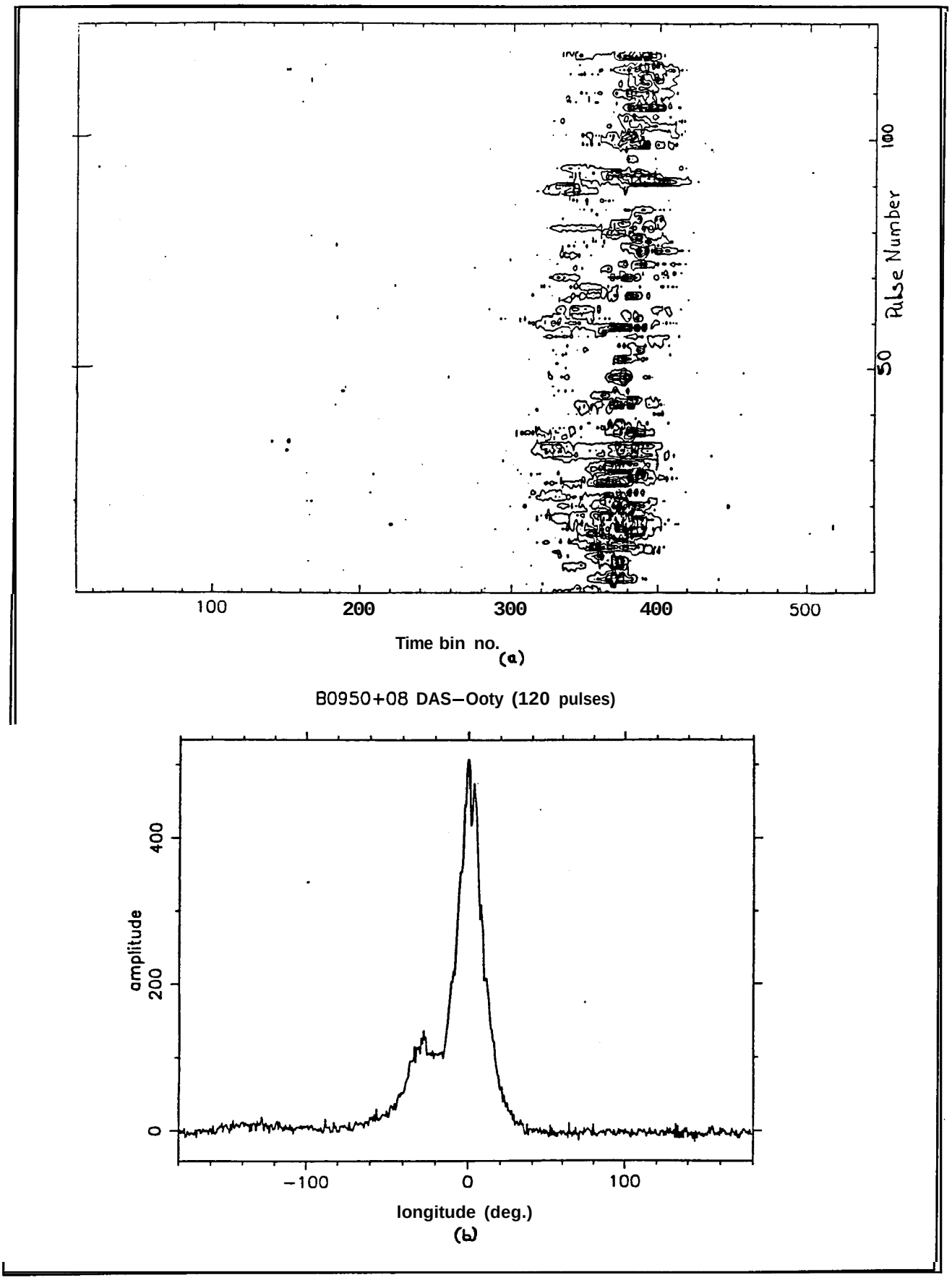


Fig. 3.18 (a) Shows a train of individual pulses from pulsar 0950+08 observed at ORT,
(b) Shows time folded profile of the pulsar