

A Low Cost Molonglo Correlator

John Bunton

CSIRO Telecommunications and Industrial Physics

21 September 2001

Introduction

One critical area of the Molonglo system is the beamformer. The resistive array is custom made and some spares are unavailable for the computer that collects the data. It would be difficult to recover from a failure in either of these components. The use of this beamformer also results in data (beams) that cannot be processed by standard methods in imaging packages such as AIPS++ or Miriad. It also removes the considerable redundancy available in the telescope data.

Advances in FPGA and RAM technology now allow a full correlator to be implemented at very low cost to replace the current beamformer. The use of a full correlator maintains redundancy and generates new data, which is currently unmeasured. This data can be processed with standard imaging packages. The redundancy and extra data will give maps with high dynamic range and lower noise. Any problems associated with processing UV visibilities from cylindrical reflector antennas can also be investigated before the building of a fully upgraded telescope.

Performance gains

Performance advantages of a full correlator are an increase in dynamic range and sensitivity. With complete correlation data the full power of techniques such as selfcal, mosaicing and clean can be applied to Molonglo data. This together with the high redundancy in the data will lead to a considerable increase in the dynamic range. An increase in dynamic range from 100:1 to 1000:1 is expected. With very accurate characterisation of the telescope dynamic ranges as high as 10,000:1 might be achieved.

Sensitivity increases because primary beam usage is increased and extra data produced. Currently the beams are formed only within the central part of the primary beam, down to the -1dB point. If mosaicing is used with the fields overlapping at the -3dB points then the number of pointings needed to cover a given area of sky is halved when compared to the current system. This will translate into a doubling of the integration time and a reduction in noise from an estimated $400\mu\text{J}$ to about $280\mu\text{J}$.

A full correlator also generates extra data because correlations between bays within an individual arm are formed. This doubles the total UV data generated. However, the data needs to be weighted to give a reasonable beam shape, which reduces the improvement. With the added data, the noise is estimated to be $240\mu\text{J}$. The main advantage is that the new spacings are offset by 3m when compared to the spacings between the arms. This gives greater UV coverage and reduces artefacts such as grating lobes. The redundancy within the data will also help with calibration.

Connection of a correlator to the existing system

The data at the output of the D/A converters that drive the resistive beamformer is both delayed and fringe stopped. Because of this, it is ready to correlate without further processing. It is a simple matter to convert the data back to a digital signal suitable for the correlator. The disadvantage of this approach is that digitising the signal will result in added noise. This is avoided if the delayed digital data is fringe stopped within the correlator. The design of the basic correlator without fringe stopping is considered first.

Correlator

A simple correlator that gives the cross power for each baseline is proposed - no spectral data. The input to the correlator is 6.25MHz complex data, which is delayed but not fringe stopped. With 88 inputs there are $88 \times 87 / 2 = 3828$ baselines to be formed but as the clock rate is low the number of complex cross-multiply accumulate (XMAC) operations needed is about 24Giga operations/sec. With FPGAs capable of 100MHz operation, this means 240 XMAC units are sufficient to process the data and with 36 XMAC units per FPGA only a small number of FPGAs are needed. With the system to be described these FPGAs are Spartan XC2S200 or smaller devices, which cost at most \$50 each (all prices Australian dollars)

The problem to be solved is maintaining the XMAC units at their full speed even though the input data rate is much lower. The solution is to double buffer the low-speed input data and process it at high speed. A suitable high-speed data rate would be about 100MHz which is 16 times the input data rate. Thus if the buffer was to hold 8ms of data it would process data for one baseline in 0.5ms. Within the 8ms it takes to fill a buffer, each XMAC unit would be able to form the correlations for 16 baselines.

The basic internal configuration of the FPGA is shown below for a 6 by 6 array of XMAC units. With two-bit complex data, there are 24 inputs into the top of the FPGA and 24 into the side inputs. The output latches allow the units to be cascaded in a two dimensional array. In the connection on the right (6 by 6 mode), all correlations are formed between the 6 antennas 1T to 6T on the top and the 6 antennas 1S to 6S on the side. If correlation between antennas within a single set of 6 is wanted identical data could be used for the top and side data but this is wasteful because each correlation is generated twice, excluding auto correlation. Only 30 useful correlations are formed. If auto correlations are generated elsewhere, say in the filterbank, the configuration on the left is possible. This configuration (dual 6 mode) generates the 15 cross correlations for two sets of 6 antennas increasing the usage of the array to 30 XMACs of the 36 available, giving an 83% usage of the available XMACs. For larger arrays the percentage of useful XMACs is even greater.

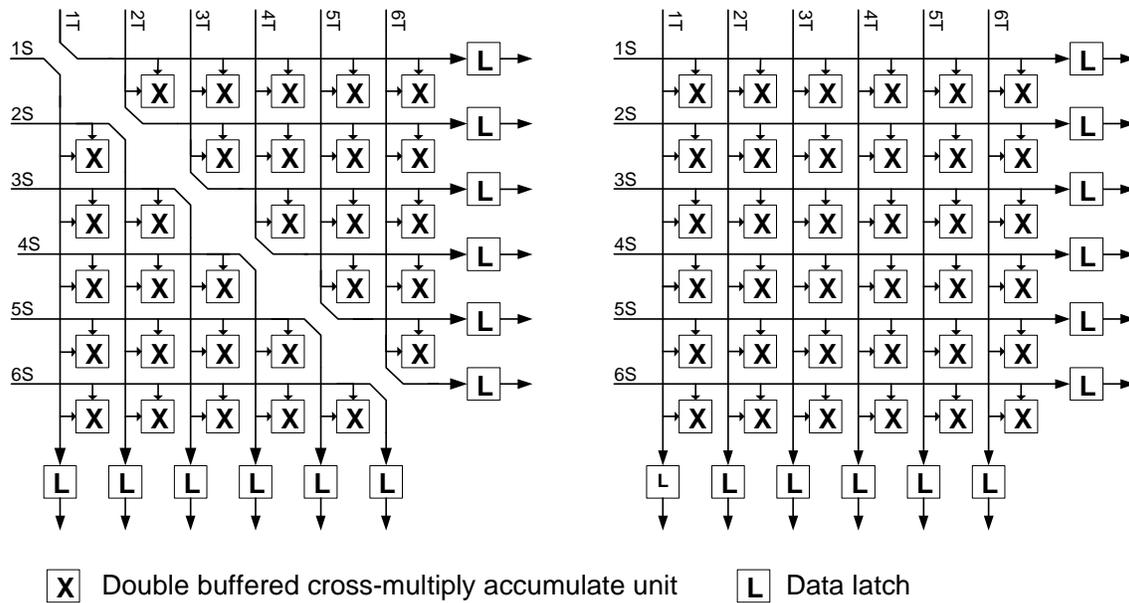


Figure 1 Example of a 36 XMAC array showing the 6-antenna by 6-antenna configurations right and the dual 6-antenna configuration left

The individual XMAC units are double buffered. While the next 0.5ms accumulations is being formed the previous result is accumulated in an external long term (4 second) accumulation memory. The buffered data for all the XMACs within a single FPGA must be stored in the external accumulation RAM in 0.5ms. Using a byte-wide memory and 32bit accumulation there are 8 reads and 8 writes per accumulation to external memory. Thus, there are $16 \times 36 = 576$ memory accesses in 0.5ms to process all the accumulation. The memory needs a cycle time of only $0.86\mu\text{s}$ to achieve this.

The data in the external memory must also be read out to the antenna control computer. This could be accomplished by alternating accumulations and reads by the antenna control computer. By double buffering the data in the external memory, the correlator can process continuously. If the XMAC performed 16 accumulations in 8ms then a 9kbyte memory is needed to store the long-term accumulation ($16 \text{ baselines per XMAC} * 36 \text{ XMACs} * 8 \text{ bytes per accumulation} * 2 \text{ for double buffering} = 9\text{kbytes} = 72\text{kbits}$). A single \$5 256kbit RAM will meet these requirements.

Size and timing of the XMAC array

With 16 cycles to form all correlations there needs to be at least $44 * 87 / 16 = 240$ XMAC units. To allow dual mode operation the array should be square. Thus, it would seem that a 16 by 16 array would be suitable. But with a basic group of 16 there are 5 and a half groups and when a half group is being processed it uses the array inefficiently. This prevents sufficient correlations being formed in the 16 cycles. One solution is to increase the clock speed to 112.5MHz and process the correlations in 18 cycles or decrease the number of groups to 5. Reducing the number of groups to 5 reduces the basic clock rate

to 81.25 MHz easing interface problems. With 5 groups the number of bays per group is 18 and to maintain the use of Spartan series FPGAs each FPGA will have a 6 by 6 array of XMACs. The correlator has a 3 by 3 array of these FPGAs providing a total of 324 XMAC units. The FPGA and accumulation memory is estimated to cost \$50. The cost of the 3 by 3 array of FPGAs and accumulation memory is estimated to be \$450.

Data ordering into the XMAC array

With 5 groups **a, b, c, d** and **e** all correlations can be formed, for example, by the following cycle of operations

| Cycle number | Top input | Side inputs | Array mode |
|--------------|-------------------|-------------|------------|
| 1 | a | b | 18 by 18 |
| 2 | a | c | 18 by 18 |
| 3 | a | d | 18 by 18 |
| 4 | a | e | 18 by 18 |
| 5 | b | c | 18 by 18 |
| 6 | b | d | 18 by 18 |
| 7 | b | e | 18 by 18 |
| 8 | c | d | 18 by 18 |
| 9 | c | e | 18 by 18 |
| 10 | d | e | 18 by 18 |
| 11 | a | b | Dual 18 |
| 12 | c | d | Dual 18 |
| 13 | Don't care | e | Dual 18 |

It is seen that group **a** is restricted to the top input and group **e** to the side inputs but all other groups must be made available to both the top and side inputs. To simplify connections into the array the buffers to the top inputs should buffer the 4 groups **a, b, c,** and **d** and the side inputs groups **b, c, d** and **e**. The total storage required for the buffers is $8\text{ms} * 6.25\text{MHz} * 72 \text{ antennas} * 4\text{bits per sample} * 2 \text{ buffers} = 3.6\text{Mbytes}$. Each group is read out 5 times but is written into the buffer once. Thus the memory data bandwidth must be 20% higher than the data bandwidth into the 18 by 18 XMAC array. A 128-bit-wide 60MHz memory provides sufficient memory bandwidth. There are 288 inputs for the data from 72 bays (4 groups) and 72 output bits to the 18 by 18 XMAC array. The memory controller will probably be implemented as a pair of controllers each processing half the antennas in each of the 4 groups, limiting the I/O pins on the memory controller to about 200 pins.

There would be 4 such memory controllers each providing half the inputs to one of the sides of the XMAC array. The controller has 2Mbytes of 64bit wide memory (or 256k x 64 bits) which can be made from 4 standard 256k x 16 memory chips. A suitable memory might be the Toshiba TC55V16256FT-15 which costs \$30 from <http://www.insight-electronics.com/cgi-bin/catalog.cgi> . Total memory cost for the 4 memory controllers is \$480. This cost can be reduced by reducing the length of the buffer or by using a unified controller that buffers both sides of the XMAC array. For any of these options the memory controllers are estimated to cost \$20. The total cost of

RAM memory and FPGAs, both XMAC and memory controller, for the full correlator is estimated at about \$1000.

Fringe stopping

The fringe stopping data is generated by the telescope control computer (TCC) every 20ms in a time multiplexed form. In each time slot a phase value and bay address is generated providing a simple interface to the correlator. This data can be used to fringe stop the data from the bays or can be applied after the correlation. Fringe stopping after the correlation avoids the requirement of data precision increase that otherwise occurs. In the current system, the phase accuracy is about 1.8 degrees. To maintain this accuracy the dual 2-bit digitised data grows to dual 7-bit data after fringe stopping. This more than trebles the amount of storage and the data interconnections within the system.

Instead of applying fringe rotation before the cross multiplication, it can be applied after. This means the application of 4005 rotations but now the input data is maintained as 2-bit complex data and the multiplication is very simple. It is also noted that the fringe rotation stays constant for 20ms. During this time, each complex product is rotated by a fixed phase and with 2-level digitisation, the complex phase term is represented by a 6 bit complex number. To maintain phase accuracy over the range of magnitudes 7 bits are needed for the real and imaginary parts of the phase rotated product. A pair of look up tables can implement fringe rotation to this accuracy, with each table containing 64 words of 7-bits or 56 LC¹s (the product of 2 3-level complex numbers requires 6 bits to encode it when expressed in real and imaginary format). If the product of the 2-bit complex product is represented as magnitude and phase then 3 bits are needed for the phase and 2 bits for the magnitude. The actual multiplication is implemented in 5 LCs. The look-up table for fringe rotation need only store the values for magnitudes 1 and $\sqrt{2}$ for the 8 distinct phases. This reduces the size of the tables to 16 words. Also as the magnitude 2 case is not stored each table is reduced 6 bits for a total of 12 LC for both tables. The magnitude 0 and 2 case is handled at the cost of a single 4 input lookup table per bit. With 7 bit output data a total of 14 LCs are needed. This approach reduces the number of LCs in the fringe rotator to 26, less than half the number needed for the direct method, and the data that has to be loaded into the lookup tables by a factor of 4.

The XMAC array processes 13 different sets of baselines every 8ms and a different fringe rotation is needed for every baseline. Thus the lookup tables must be updated every 615 μ s. The actual data to be loaded can be stored in the FPGA Block RAM but some time and extra circuitry will be needed to load the data. With 72 6-bit tables to be loaded, allowing 36 μ s to load the tables leaves 500ns to load the 16 values. Breaking the XMAC array up into 4 groups of 9 each with separate load circuitry increases the time to 2 μ s which is easily achievable. Implementing the look up tables as dual ported memory provides a simple method for loading the data. This increases the cost of the fringe rotator to 38 LCs per XMAC. Most of the logic needed for the fringe rotation can share the same LCs as the memory needed for the double buffer. Thus if dual 16 bit accumulations are used an extra 4 LCs are needed to implement the logic for the fringe rotator. Add the 5 LCs needed for the multiplication and the number of LCs needed for an XMAC is 73 or

¹ Here an LC (logic cell) is defined to consist of look-up table (LUT) and a register

2628 LCs for a 6 by 6 XMAC array. The major cost in this approach is an increase in the clock rate to 86MHz and a need to buffer the antenna data.

Wide field operating modes

The correlator can only process data at the full bay level. This limits the field of view in meridian distance to just over 1 degree (-3dB). For large field mapping it will be necessary to form correlations for two or three pointings and the data processed as a mosaic. With sufficient attention to detail, a dynamic range of over 1,000 to 1 is achievable with selfcal. For image production on site we might keep the current system and just transform the correlation data to form a strip scan which can then be processed with the current software.

A second approach to widefield imaging is to disable half or three quarters of each bay. This results in a more circular beam. With only a quarter of each bay active the instantaneous sensitivity is reduced by a factor of 4 but because there is no beam switching the actual loss is a factor of 2. This approach should allow nearly 4-degree fields to be imaged with very straightforward processing.

Another interesting option is to operate some bays with half or a quarter of the bay active. With a third of the telescope operating with a half bay active and a third operating with one quarter active the effective area is reduced to 60% of the full area. For wide field imaging the sensitivity loss is about 20%. The advantage of this approach is that nearly complete UV coverage is achieved for each of the three sets of bays, full, half and quarter. In addition, there are the correlations between the sets giving data at 6 different beam widths. For the half bays, there are two possible choices as to which half of the bay to use. For the quarter bays, there are four. Using a mixture of these choices gives an increase in the number of base lines and better sampling of the UV plane. If the complexities of imaging with this data can be handled then the increased richness of the data will lead to improved wide-field image quality.

Conclusion

The hardware cost of a full correlator for the current Molonglo 3MHz-bandwidth 88-bay telescope is small. Upgrading to this correlator improves the reliability of the telescope, provides increased sensitivity and dynamic range and explores the challenge of processing correlation data from a telescope based on a cylindrical reflector. The major challenge is to find the manpower to design and build the hardware, integrate it into the current telescope and develop the necessary processing to give high dynamic range images.

Acknowledgement

The authour would like to thank Colin Jacka for his critical comments on this paper.