DELAY & DPC CONTROL CARD (ver 1)

APRIL 17, 1994

Alka Dikshit

1 Introduction

This card has been designed as the *prototype for final Control Card* for DPC cards and DELAY cards of the final system. The wired prototype is ready and working. The card is a 3U card (so as to sit in the same bin with DELAY and DPC cards which are also 3U.

This control card *now onwards referred to as control card* was required to provide all the control signals necessary for the DPC card, all control signals for the DELAY card as well as the delay values for each DELAY card in the bin in each FFT cycle. The card is designed such that it will cater to the needs of six DPC cards and twelve DELAY cards. Besides providing control signals this card also reads Total power from DPC cards and distributes Walsh functions for demodulation to DPC cards.

1.1 Specifications of The card

The following are the basic functions which the card should be able to handle.

* This card should be able to communicate asynchronously with the rack control card without any bit errors. So that the required settings for control lines and delays can be communicated by Rack control card, and total power with a time stamp in terms of system clock and FFT cycle number of a STA cycle can be communicated back to rack control card.

* Should be able to provide 11 bit delay to each DELAY card in the bin in one FFT cycle, and should be able to communicate 2 bits of control to DELAY cards.

* Should be able to provide DPC control word setting every STA cycle. *Although the control word setting for DPC will be only done in the start of scan by master control card, just for the sake of redundancy and as a precaution against noisy environment the DPC control word is set every STA cycle.*

* Should be able to read the power accumulated for each channel on DPC card at least once in two FFT cycles.

* Should be able to transfer the read powers for all channels after accumulating it further.

* should be able to read walsh pattern

* Should be able to provide walsh demodulation patterns to all DPC cards in a bin.

* Should be able to provide test patterns *as downloaded by the master control card at the start of session* if required to specified DPC card.

* Should be able to recognize and communicate which card it has to read or write to.

1.2 Design Guidelines

Before going in to the electronic aspects of the card we will like to specify the design strategy chosen, and the reason for the same. The card is designed to fulfill all the requirements mentioned above for a bin i.e. six DPC cards and 12 DELAY cards.

Looking at the functions this card has to perform in the timescales given, it is best to use a computation intensive processor as the core of the card. Hence a DSP (digital signal processor) AD2105 was chosen as the core of the card. Each delay and DPC card is ports mapped on this processors memory. Later it was found that the DSP working at 10 MHz can't take care of so many functions because the overheads for house keeping are too many *as it is not a controller* . Besides this as the DSP processor works at 100 ns instruction cycle rate the skew between two walsh may become too much. Thus the function of distributing walsh functions to DPC's was decoupled from the DSP processor.

As DPC cards were already designed to receive the walsh patterns on DSP bus. This was decided to use the BUS request and grant facilities with the processor to distribute walsh patterns as and when required. The processor allows bus control for outsiders on request. To reduce the skew between different walsh it was decided that the walsh distribution will be done by a STATE MACHINE working at 32 Mhz. instead of involving processor in this work. This is presently not implemented in the prototype because of various reasons.

As the test data is to be provided at 32 MHz it is best implemented as a RAM which is written at the start of the session at the speed of DSP instruction cycle and later read at 32 MHz cyclically. This RAM has to be at least 512 words long as the DELAY INIT is also generated on the same card and one FFT cycle is 512 cycles of the slow (32 MHz clock). Delay card dual port ram 4K data deep hence it is decided to put a 8K deep RAM on the card. (present prototype has 2 K deep RAMs.

As this DSP provides memory mapped I/O , it is decided to have each DPC and DELAY card to be a I/O port for the processor and read and write to respective addresses, for various control signals and delay which go on DSP bus on backplane.

For the power reading and timing it is decided to have a port 16 bit wide *DSP processor allows two kind of ports 24 bit (program memory mapped) and 16 bit (data memory mapped).* So if DSP wants to read power it reads this port which has 16 bits of power from DPC (two channels) and in the next cycle it reads the second port for time.

There is decided to be a status port which sets control signals specific to the control card itself.

This completes the basic design philosophy of the card.

The control card is divided into the following sections.

- Input/output section.
- Processor its memory interface & communication interface.
- Memory mapped I/O decoding.
- Test pattern, init pulses & interrupt generation.
- Control signals & Clock section.
- State Machine to distribute walsh patterns fast on DSP bus.

The next section will describe the card electronics as per the prototype.

2 Input/Output Section

The input to this card is the clock of 32 MHz, The control and data lines of centronics parallel port, STAinit *(see appendix 1)*, and walsh pattern for all the DPC cards in a bin.

The output of the card is the 16 bit wide bus of DSP processor, walsh pattern 4 bit wide bus *it is the mux part of DSP bus this way walsh remains independent of the DPC control settings.*, test pattern 12 bit wide bus, Card select lines for all the cards in the bin except control card and clock card, and two more control lines for function selection on cards, delay load pulse , powerlatching pulse,and DSP bus read and write signals.

The output signals are on a separate 4 row EURO connector to facilitate its sitting in the back plane. The input comes on a FRC connector from the front of the card. Although the clock and STAinit comes from the backplane.

There are transreceiver buffers and latches to facilitate the bus diving on the backplane at the output section.

3 Processor its Memory Interface & Communication Interface

This section is based on AD2101 DSP processor. The Data memory and program memory are 4Kx23 each. Besides there is a BOOT ROM of 8Kx8 size. The address bus is buffered. Data bus uses bidirectional buffers whose direction depends on the read pulses from the processor. The processor is Booted with the power on reset. But a reset is provided on

board to facilitate intermediate reset if required. The centronics data bus is latched at a memory mapped I/O. for the middle 8 bits of the processor bus. The control lines are latched to the upper 8 bits. *This section will finally be replaced by a full duplex serial link, though the processor will still see the parallel interface. This will be done with the INMOS transputer to parallel processor bus interface IC, C012 because the Rack control card is a transputer based card.*

4 Memory mapped I/O Decoding

4.1

The entire data memory address space of DSP 2101 is first divided in to blocks of 512 words with the help of 3x8 decoder. These blocks are then further decoded in to blocks of 64 words with the help of two tailor made 7x8 decoders.

4.2

This section has been changed later to decode half of the Data memory and half of the program memory decoded in two 512 word blocks. *This will facilitate having access to 24 bit wide I/O ports if required.*

4.3

The program memory 64 word space is then divided to generate DPC and DELAY ports. The details of these ports are available in the architecture file of ADSP2105 for this card.

4.4

The data memory 64 words space is divided to generate status port power port, communication port, time port etc; which require 16 bit wide bus. Details are in the architecture file of the ADSP2105.

5 Test Pattern, Init & Interrruprt generation

5.1 Test RAM address generation

There are two RAM of *2Kx8 each now but to be replaced by 8Kx8 each in the final card* which distribute 12 bit test data to the specified DPC card The RAMs are written at

the startof the session. The required RAM is to work at processor write cycle in write mode, and 32 MHz in read mode. Besides both read and write should be sequential and predetermined. To achieve this the address to RAM is generated with the help of a 11 bit counter, and the clock to the counter is controlled according to read or write function. The clock to the counter is processor write pulse if the cared control indicates that it is in testdata writing mode else it is 32 MHz in normal mode. The test data is read from RAMs all the time as delayinits are also generated on the same RAMs but the test data is disabled to go on the backplane bus unless request for sending test data is received. This is done to reduce noise on the backplane.

5.2 Init & Interrupt Generation

The counter could have been enabled only when the test data is required instead of letting it free running to save energy and heat dissipation. This can't be done though it is a good practice, because there are 4 bits extra in the RAMs which are used to generate Dlyinit *(see appendix 1)* and the interrupt for the processor. Hence this information also should be downloaded with the test pattern.

These inits and interrupts are then synchronized with the 32 MHz system clock and distributed on the control card and backplane to be tapped by different cards as per requirement.

6 Control & Clock Section

This section deals with the processing of clock after it enters the card and the other control line .

6.1 Clock

The Control card has two clocks a. One is 10 MHz clock for the DSP. b. Other is 32 MHz system clock.

The 10 Mhz clock is generated on board with the help of a crystal oscillator. Which only feeds the DSP. The 32 MHz clock enters the card via backplane as a TTL signal. This clock is used to clock the latches for the testdata after inversion, To make sure that the DPC card which will latch this data will get stable data when its clock is rising. This clock passing through some logic is also provided to the counter for address of test data RAMs. This clock is also used by the State Machine for its synchronous part.

6.2 Control

The control section basically comprise of a onboard status port and some combinatorial logic to manage various buses on the card and various modes of card operation. Some other specific ports also work as the I/O mapped memory on the card.

This card has some LEDs to indicate the status of processor at various points which are listed in the Schematic.

7 State Machine for Walsh Distribution

The state machine is a Mealy machine. Its has six well defined states. It takes as input the Walsh pattern serially shifted in. A counter will keep track of the states and next state. The transition from one state to other will take place depending upon the Walsh pattern.

The walsh pattern will be tested for a change in value *(with respect to the previous value)* This can very trivially be done with the help of XOR gates. And then enter the State Machine after a bus grant is provided by the DSP processor, i.e. the bus grant signal from DSP will initiates the State Machine and resets the Machine's internal counter.The new walsh pattern is loaded and shifted out of a serial register one by one. At each clock cycle a counter gets updated which, together with the shift register output, will define in which state the Machine will go in. In a maximum of 6 clock cycles /it besides the time it takes for the processor to release its bus . (after it detects a change in walsh pattern) it will be able to set the new walsh patterns on relevant DPC cards . It will leave the DPC cards which don't undergo a walsh change untouched which will avoid any possibility of glitches on those DPC's.

The State Diagram is as shown. This require a altera EM 5032 EPLD a 22V10 and some more discrete logic *this is optional.* The 22V10 will be programmed as a Mux to allow State Machine generated signals on DSP bus when DSP asserts a bus grant. The EM5032 EPLD will hold the State Machine and the shift register (hopefully, if not that has to be put separately.). and discrete logic will take care of the XOR gates.

8 Appendix 1

- FFTINT: This pulse is the indication for the start of a FFT cycle. This is equal to 512 clock cycles of system clock. After this there are four clock cycles of dead time (this is explained below). Hence the total gap between two FFT cycles is 516.

- Dead Time: These are four clock cycles between end of one FFT cycle and start of another. The dead time is called so because in this time the FFT engine is

virtually dead. This is required because first 4 point FFT can be calculated once 4 inputs are already read in.

- STAINIT: This is another synchronizing pulse used in the correlator system. This pulse can come every X FFT cycles (where X is some multiple of 256 FFT cycles) This pulse is also used to reset write counter on DPC. This is done just to ensure recheck on the system .This is synchronized with FFTINT synchronization is required because the read counter on DPC is reset every FFT cycle and the instant at which read counter is reset ,should be the same when write counter is reset and the proper relation is maintained.

- Dlyinit: This is the pulse exactly same as FFTinit but is used in presenting DELAY card read counters for the dual port RAMs.

STATE DIAGRAM FOR WALSH DISTRIBUTION