Null Port Controller Design

Daniel Gordon

February 11, 1993

1 The Null Port Controller

The null port controller is a very simple version of the port controller. It has no processor and does only VCI mapping and FIFO queuing. A Xilinx chip controls all its functions. It uses triple ported DRAMs to look up the new VCI, and a FIFO to do speed matching with the transmission board. Two latches, a hex inverter and a Xilinx PROM are the only other chips on the board. Figure 1 shows an overview of the null port controller.

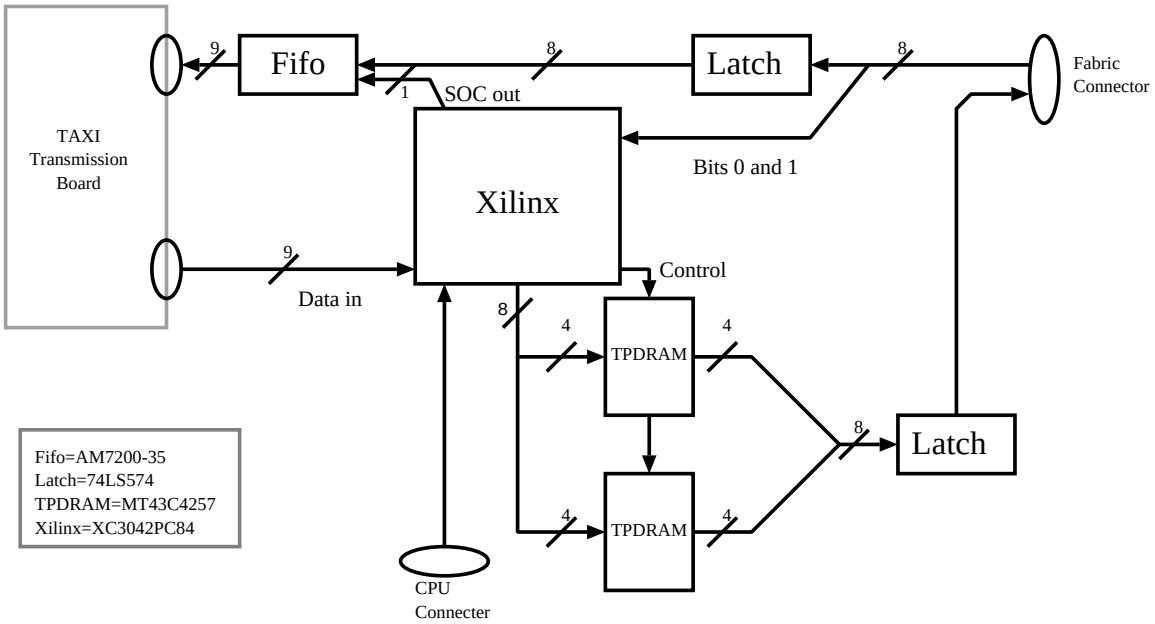

Figure 1: Null Port controller layout

The Null Port Controller controls the flow of data in both directions from the switch fabric to the transmission interface and vice versa. The fabric-to-transmission part is simple. When the Xilinx chip detects bit zero set from the fabric it reads the 53 bytes which follow it into the FIFO. The first byte read (the port controller routeing byte) then becomes the byte with SOC set. If bit one in the port controller routeing byte is set then the cell is a loopback cell indicating that it should be sent into the transmission input side of the null port controller. On a Fairisle port controller this is used to send cells to the processor. This is meaningless on the null port controller, so cells with this bit set are dumped.

The transmission-to-fabric section waits for 'frameIn' from the fabric and then tests the FIFO on the transmission board to see if it 'not empty'. If it is then the data is read out, and the 2 bytes after SOC (the VCI) are latched and used to set the memory address for the rest of the cell to be written to. At the next 'frameIn', the cell is read out with the new VCI and routeing bytes prepended to it and sent into the fabric. (See section 3)

2 The Xilinx Chip

The Xilinx chip is an XC3042PC84-100 logic cell array. It is programmed by a XC1736A Xilinx PROM on power-up. Its main function is to store a cell (received from the transmission interface) in the correct part of the memory, read the cell back and send it to the fabric, and keep the memory refreshed. It also controls the data coming back from the fabric (placing it into a FIFO) and creates a simple link between the CPU interface and the DRAM access to the memory. Table 1 shows the 69 of its 84 pins which are used for signals. Figure 2 shows a schematic diagram of the functional parts of the Xilinx chip

Purpose	Quantity
FrameIn, FrameOut, Phi1 and Phi2	4
Data from transmission interface	9
SOC to transmission interface	1
Control lines to tramsmision interface	4
Data to the fabric and Ack	9
Least significant bits of data from fabric and Ack	3
Write enable for latch before fabric	1
Address lines to the memory	9
Data lines to the memory	8
Control lines to the memory	9
Data lines for the CPU interface	8
Address lines for the CPU interface	2
Control lines for the CPU interface	3

Table 1: Pin usage on the Xilinx chip

3 The Memory System

The Triple Ported DRAMs (TPDRAMs) on the null port controller are 2 512x512x4 bit memory chips. They are addressed via a 9 bit column address and a 9 bit row address. They can be accessed in three ways, two serial and one parallel. The parallel access is used to load the memory with routing and VCI information (via the CPU interface), and the serial accesses are used to write and read a cell to and from the memory. The memory in logically divided into 8x512x64 byte blocks, any of which can hold one cell. When a cell arrives on the transmission side of the null port controller, the 2 byte VCI is extracted and converted into a row and column address (see table 2). This gives the address of a 64 byte block into which the cell can be put.

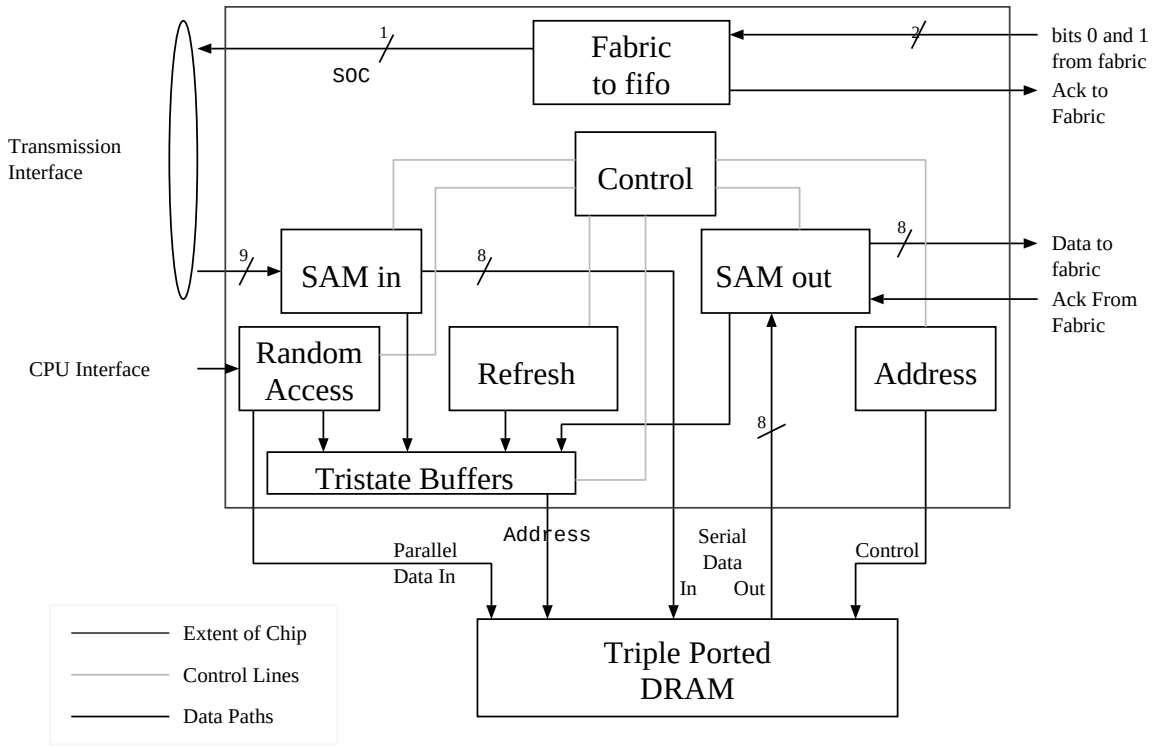

Figure 2: The Xilinx layout

VCI 0 bit	0	1	2	3	4	5	6	7
	r1	r2	r3	r4	r5	r6	r7	r8
VCI 1 bit	0	1	2	3	4	5	6	7
	-	-	-	-	c6	c7	c8	r0

Table 2: VCI to memory location convertion

Figure 3 shows how the cell is placed in the 64 byte memory block. Once the VCI has been removed the other 50 bytes (2 FAS and 48 data) are written into the memory starting at byte 4. This is done using one of the serial access memories (SAMs) in the TPDRAMs [1]. The 50 bytes are shifted into SAM(b) [2] starting at the column address (which has the value 64n+4). At the same time 50 1's are shifted into a bit mask registers (at the same column address). The data in SAM(b) is transfered to the row in memory specified by the row address, but only where the bit mask register has ones. At the next 'frameIn' signal the data is read out of the same block, but this time 54 bytes starting from byte 0. The row in memory with the cell in it is transfered to SAM(a) and shifted out (starting at column 64n) to the fabric via a latch. The routeing information and the new VCI are thus prepended onto the start of the cell.

[1] These are logically 8 bits wide, though they are divided between the two TPDRAMs

[2] There are 2 SAMs labelled (a) and (b)

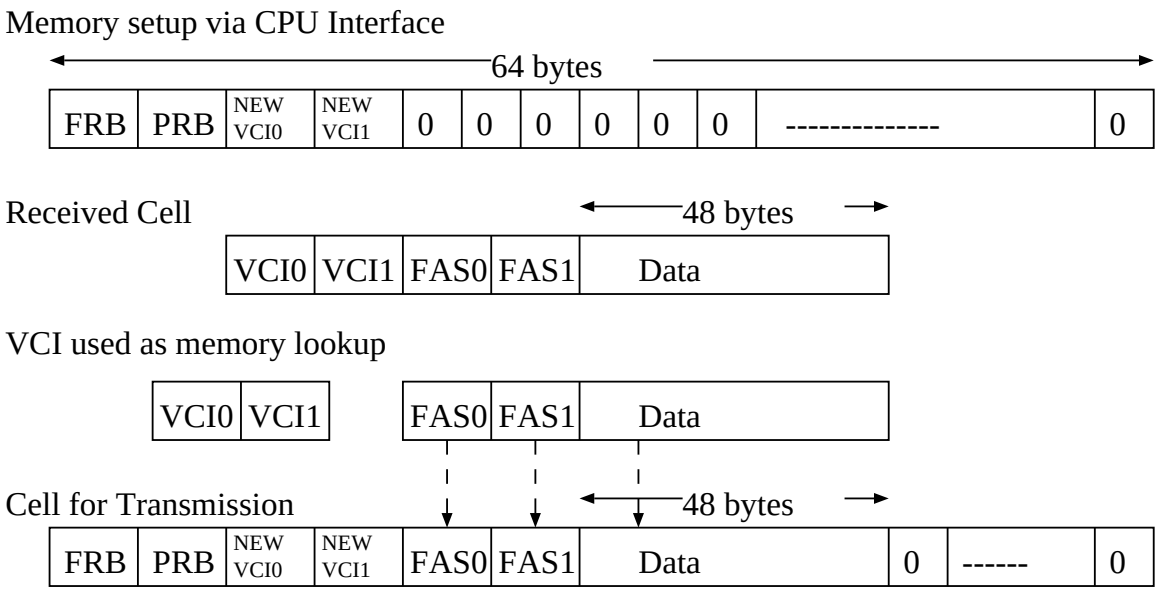

Figure 3: Cell storage

3.1 The Xilinx Registers

There are four registers on the Xilinx chip (set via the CPU interface) which are used to setup the address and data for a DRAM write into memory. Table 3 shows the usage of these registers. Data is written into memory one byte at a time by writing the column address into c[0..8] and the row address into r[0..8]. When the data is written into d[0..7], the Xilinx chip does the write into memory and pulls 'Rdy' low when it has finished. On reset, both input and 'send zero' are disabled. Two flags SZ (send zero) and ID (input

Address	bit	0	1	2	3	4	5	6	7
0		c0	c1	c2	c3	c4	c5	c6	c7
1		c8	r0	r1	r2	r3	r4	r5	r6
2		r7	r8	SZ	ID	-	-	-	-
3		d0	d1	d2	d3	d4	d5	d6	d7

Table 3: Registers on Xilinx Chip

disable) are set and reset by writing to register 2. The 'input disable' flag prevents data from the transmission interface from being written into the memory. This is used to stop cells being VCI mapped before the new VCIs and routing information has been written into the memory. The 'send zero' flag is used for debugging. When it is set, the null port controller will send the cell stored in row 0, columns 0 to 63 to the fabric every time a 'frameIn' signal is received.

4 The CPU Interface

Data is written into the Xilinx registers using an Archimedes simple podule. This is plugged into the podule bus of a port controller and connected by ribbon cable to the null

port controller. The registers on the Xilinx chip are thereby memory-mapped into the podule I/O space of the Archimedes/port controller (See table 4).

Up to 8 null port controllers can be connected to one podule. In order to select which device the podule is to write to, 8 signals are sent to each board. One of eight links is set on each board to specify whether that board should respond to the current write request.

Podule Bus	A4	A3	A2	D15 - D8	D7 - D0
Null Port Controller	Strb	A1	A0	D7 - D0	S7 - S0

Table 4: Memory Mapping

A write to a register is performed by writing to the register address times 4 (plus the podule base address) and then to this location plus 16 which sets 'Strb' to high. The rising edge of 'Strb' latches the data into the appropriate Xilinx register. D[0..7] becomes the data for that register and S[0..7] are 8 select lines. Reading from any location in the podule space will give the podule ID (See table 5). This is set to indicate that the podule does create interrupts and is Acorn complient. It also has a 4-bit identifier, the least significant bit of which is used as a ready signal from the Xilinx chip to indicate that the data has been written into memory.

Bits	Use	Meaning if set to 1	Value
0	IRQ	Requesting an IRQ	0
1	P	Podule is not present	0
2	FIQ	Requesting a FIQ	0
3-6	Podule ID	N/A	see text
7	Acorn	Not Acorn complient	0

Table 5: Podule Identifier Byte

Writing a byte into memory might consist of:

1. Write to PBA[3]+0
2. Write to PBA+0+16
3. Write to PBA+4
4. Write to PBA+4+16
5. Write to PBA+8
6. Write to PBA+8+16
7. Write data to PBA+12
8. Write data to PBA+12+16

[3]Podule Base Address.

9. Read from PBA until 'ready' becomes low

Both the simple podule and the null port controller have IDC 26-way sockets on them into which a ribbon cable is connected. Table 6 shows the pinout.

Pin No.	1	3	5	7	9	11	13	15	17	19	21	23	25
Use	Gnd	Gnd	Gnd	Gnd	Gnd	S0	S1	S2	S3	S4	S5	S6	S7
Use	Strb	Rdy	-	A0	A1	D0	D1	D2	D3	D4	D5	D6	D7
Pin No.	2	4	6	8	10	12	14	16	18	20	22	24	26

Table 6: IDC 26-Way connector