Version 1.2

OC
Output Controller

Implemented in the
ROD Demonstrator Board

Daniel La Marra

1. Index

1.1 Chapters

1.2 Tables and Figures

2. Description

The main task of the Output Controller (**OC**) is to manage and control the data flow from the four Processing Unit (**PU**) Boards to the ROB (**Read-Out Buffer**) via the link.
It must be able to transfer, when requested by the user, one event fragment to the output memory, which can be read by a computer through the VME bus. In this case the memory is used as a spy memory. If the link is not available the Output Controller is able to transfer the entire event fragment to the output memory. In this case the output memory is used as a buffer memory. This is the mode of operation in the test beam. The crate computer will read the buffer memory at the end of each burst. The memory size is 16 Mbytes.

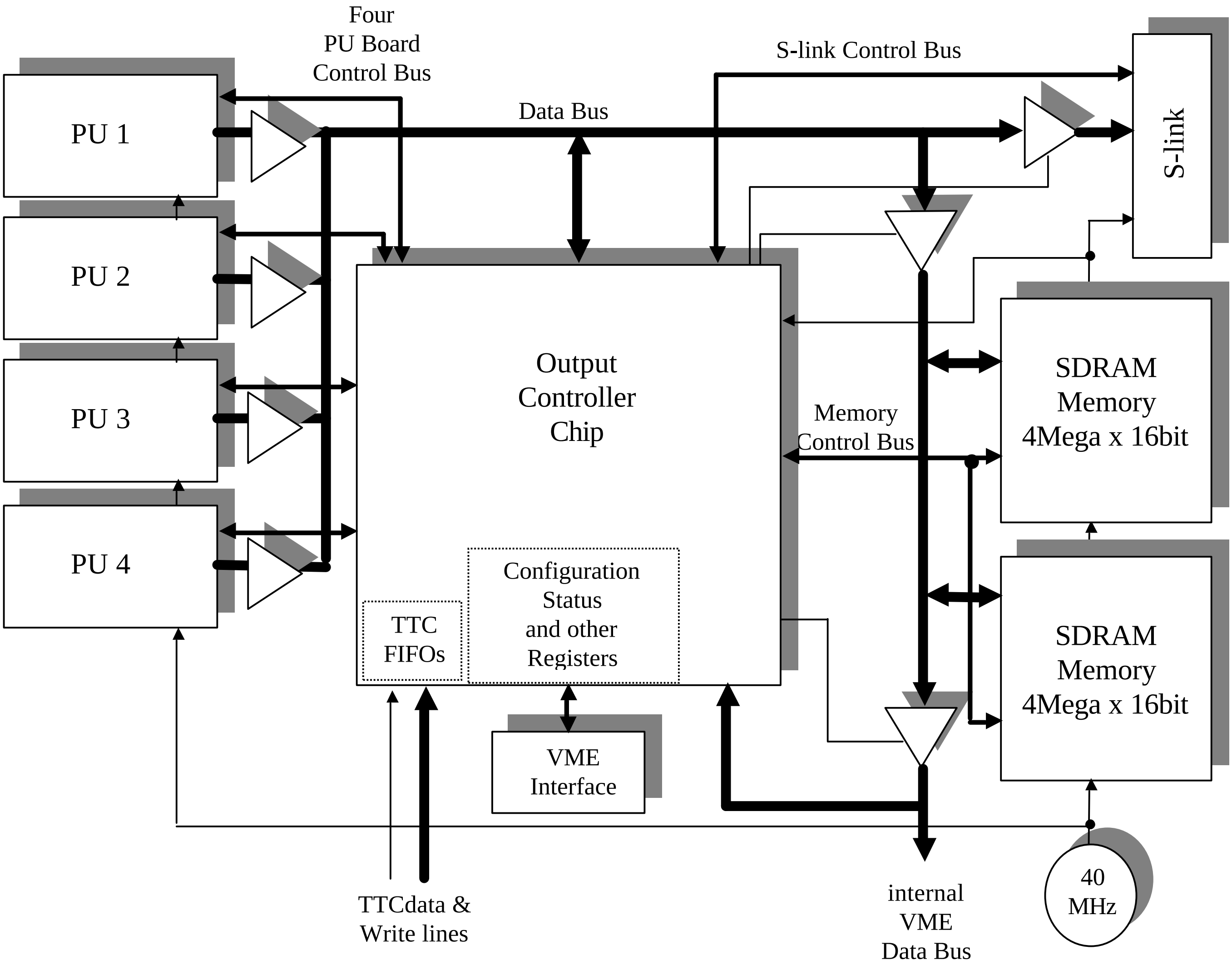
3. Block Diagram and bus definition

3.1 Block Diagram

The block diagram below shows how the Read Out Controller is connected to the other elements. It must be able to manage these three kinds of control bus:

- Four PU Board Control Bus
- S-link Control Bus
- Memory Control Bus

Figure 3.1—1 : Block Diagram



3.2 Four PU Board Control Bus

A 5-lines bus controls the data output of each PU board:

Table 3.2—1 : PU Board Control Bus

Name	Origin	Function
fifo_evt_rdy	PU board	When high, indicates that the PU Board has one complete event.
fifo_rdclk	OC	Read synchronous clock input. 40MHz.
fifo_oen	OC	When low, the 32-bit output is enabled. When high, the output bus must be in high impedance state.
fifo_rdenn	OC	When low indicates to the PU Board that it must give the next data on the bus, at least 10ns after the low-to-high transition of the fifo_rdclk signal.
fifo_evt_end	OC	This 25ns positive width signal indicates to the PU Board that the entire event is read.

3.3 S-link Control Bus

This bus allows the OC to control the data, which is sent to the ROB via the S-link. It is described by the CERN, ECP Division in the document named:

The S_LINK Interface Specification.

The postscript version is on the Web at:

http://www.cern.ch/hsi/s-link/spec/spec/s-link.ps

There are twelve lines, which are described in the table below:

Table 3.3—1 : S-link Control Bus

Name	Origin	Function
uwen_b	OC	User Write Enable. When low enables data to be transferred to the S-link on the low-to-high transition of the clock.
uctrl_b	OC	User Control line. When low indicates that the data transmitted is a control word.
udw<1:0>	Not Used	User Data Width lines. Define the data width the S-link is to be operated in. These two lines are connected to the ground, which defines 32-bit width.
utest_b	OC	User Test line. When low switches the S-link in test mode.
ureset_b	OC	User Reset line. When low initiates a reset cycle.
ldown_b	S-link	Link Down. When low indicates that the S-link is not operational.
lff_b	S-link	Link Full Flag. After it goes low, up to two more words may be written.
lrl<3:0>	Not Used	Driven by the S-link. Link Return lines.

3.4 Memory Control Bus

The memory chip chosen is the MT48LC4M16A2-10 from Micron, which is a Synchronous Dynamic RAM with a configuration of 4 Meg x 16 bit. Only two chips are needed to reach 16Mbytes.
The price for very low quantity is about 27 CHF each.

The data sheet is on the Web in PDF at:

http://www.micron.com/mti/msp/pdf/datasheets/64MSDRAM.pdf

Twenty-one lines are needed to control the memory. These lines are shown in the table below:

Table 3.4—1 : Memory Control Bus

Name	Origin	Function
cke	OC	Clock enable. When high the clock input on the memory chip is enabled.
cs_b	OC	Chip select. When low the memory's command decoder is enabled.
cas_b, ras_b, we_b	OC	Command input. These three lines define the command being entered.
dqm<1:0>	Not Used	These two lines are connected to the ground, which allows 16-bit bus width on each memory chip.
addr<11:0>	OC	These lines can be the Row address (addr<11:0>) or the Column address (addr<9:0>) depending of the command.
count_bank<1:0>	OC	These two lines are used to select the bank. There are 4 banks in each chip. Each bank is 1Meg x 16 bit.

4. VME Interface

This part of the Output Controller allows the CPU to write and/or read in several 32 bit registers, through the ROD Mother Board VME Interface. The CPU can:

- Read and write the Configuration Register.
- Read the Status Register.
- Write the Format Version Number.
- Write the Source Identifier
- Write the Detector Event Type
- Write the Detector Format Version Number (16 bit)
- Write the Reset Register.
- Read the SDRAM in the VME sequential mode

4.1 Configuration Register

This 32-bit read-write register is loaded by the VME interface from the VME bus:

Table 4.1—1 : Configuration 32 bits Register Pattern

Configuration Register.							
31 (MSB)	30	29	28	27	26	25	24
reserved	reserved	reserved	reserved	en_tst_link	dtm	en_spy	en_link
23	22	21	20	19	18	17	16
mask8*	mask7*	mask6*	mask5*	mask4	mask3	mask2	mask1
15	14	13	12	11	10	9	8
reserved	reserved	bankmax1	bankmax0	rowmax11	rowmax10	rowmax9	rowmax8
7	6	5	4	3	2	1	0 (LSB)
rowmax7	rowmax6	rowmax5	rowmax4	rowmax3	rowmax2	rowmax1	rowmax0

* Not used in the ROD Demonstrator Board, used only in the final version of the ROD.

Table 4.1—2 : Configuration Register Contents

Bit	Name	Function
<13:0>	bankmax<1:0> and rowmax <11:0>	Used only when the S-link is disable. Therefore when the SDRAM is enabled. This pattern allow the Output Controller to stop the storage when this memory location is reached. The Output Controller stops only when a whole event is stored. (the SDRAM 's 8 bit column address are not checked)
<15:14>	reserved	
<23:16>	Mask <8:1>	Mask PU board. When a bit is set the corresponding PU board will be masked. If a PU board is masked the Output Controller will not read it.
<24>	EnLink	Enable the S-link. When this bit is set, the data flow will be sent to the S-link, Otherwise it will be stored in the memory (SDRAM).
<25>	EnSpy	Enable Spy Channel. When EnLink is set, this bit allows reading one event on the VME bus through the SDRAM memory. If EnLink is low this bit has no effect.
<26>	dtm	Data Taking Mode. Used only when the S-link is disable. Therefore when the SDRAM is enabled. When is set its allows the Output Controller to start the storage of events. When is reset its allows the Output Controller to stop the storage of events only when a whole event is complete.
<27>	EnTstLink	Enable Test Link. When EnTstLink is set, the line utest_b, in the S-link Control Bus, is activated. This allows putting the S-link in test mode.
<31:28>	reserved	

4.2 Status Register

This 32-bit read only register give information which can be transferred to a CPU through VME bus:

Table 4.2—1 : Status 32 bits Register Pattern

Status Register.							
31 (MSB)	30	29	28	27	26	25	24
reserved	headfifo_empty	TTCfifo_empty	header_ready	reserved	e0e_flag	link_fail	mem_complete
23	22	21	20	19	18	17	16
mem_empty	data_avail	mem_size21	mem_size20	mem_size19	mem_size18	mem_size17	mem_size16
15	14	13	12	11	10	9	8
mem_size15	mem_size14	mem_size13	mem_size12	mem_size11	mem_size10	mem_size9	mem_size8
7	6	5	4	3	2	1	0 (LSB)
mem_size7	mem_size6	mem_size5	mem_size4	mem_size3	mem_size2	mem_size1	mem_size0

Table 4.2—2 : Status Register Contents

Bit	Name	Function
<21:0>	mem_size<21:0>	This pattern is the write pointer address. It can be used as a word count for the DMA VME burst read access.
<22>	data_avail	Data available. When is high it indicates that the memory's data can be read. This flag generates a interrupt on the VME bus.
<23>	mem_empty	Memory empty. When is high it indicates that the memory (SDRAM) is empty.
<24>	mem_complete	Memory complete. When is high it indicates that the memory as reached the bankmax and rowmax of the configuration register.
<25>	link_fail	Link failure. When is high it indicates that after the reset the "ldown" signal of the SLINK interface, stay low. The SLINK will be disable.
<26>	e0e_flag	End Of Event flag. When is high it indicates that the Output Controller did not find the End Of Event pattern as the last word of one or more event in one or more PU. This flag can be reset through the Reset register. This flag generates a interrupt on the VME bus.
<27>	reserved	
<28>	header_ready	Header is Ready. When is high it indicates that the internal Header Builder has build the header. So the event can be transferred since the Header FIFO is ready.
<29>	TTCfifo_empty	TTC input FIFO is Empty. When is high it indicates that the internal input TTC FIFO is empty. If it is low it indicates that at least one event is pending.
<30>	headfifo_empty	Header FIFO is Empty. When is high it indicates that the internal Header FIFO is empty. So if there is an event in the TTC input FIFO the Header FIFO must be loaded by the Header Builder.
<31>	reserved	

4.3 Format Version Number Register

This register is write only and it must be loaded with the Format Version Number, which is a 32-bit integer number, and it defines the version of the ROD data fragment. This number is transferred to the header of the Event Fragment.

4.4 Source Identifier Register

This register is write only and it must be loaded with the Source Identifier, which is the 32-bit word that defines the fragment. This number is transferred to the header of the Event Fragment.

4.5 Detector Event Type Register

This register is write only and it must be loaded with the Detector Event Type, which is a 32-bit word. It is a sub-detector specific event type flag. This number is transferred to the header of the Event Fragment.

4.6 Detector Format Version Number Register

This register is write only and it must be loaded with the Detector Format Version Number, which is a 16-bit word that indicates the format version for the detector data block. This number is transferred to the status word of the Event Fragment

4.7 Reset Register

This 32-bit write only register is used by the CPU, through the VME interface, to reset independently some parts of the Output Controller. The table below shows the affectation of each bit:

Table 4.7—1 : Reset 32 bits Register Pattern

Reset Register.							
31 (MSB)	30	29	28	27	26	25	24
not used	not used	not used	not used	not used	not used	not used	not used
23	22	21	20	19	18	17	16
not used	not used	not used	not used	not used	not used	not used	not used
15	14	13	12	11	10	9	8
not used	not used	not used	not used	not used	not used	not used	not used
7	6	5	4	3	2	1	0 (LSB)
not used	not used	not used	not used	not used	rst_general	rst_EOEflag	rst_link

Table 4.7—2 : Reset Register Contents

Bit	Name	Function
<0>	rst_link	Reset S-link. When this bit is written to one, the line ureset_b of the S-link Control Bus will be activate and the S-link reset cycle will be initiate.
<1>	rst_EOEflag	Reset the End Of Event flag. When this bit is written to one, the End Of Event flag will be reset, and therefore it releases the interrupt line. This flag can be read in the Status Register.
<2>	rst_general	Reset the whole OC Chip. It is like the external hardware reset except for the registers which can be written from the VME bus. These registers will be not changed by this reset.
<31:3>	not used	

5. Interruption

5.1 Interruption Description

One interrupt line is sent to the ROD Mother Board VME Interface. Two events, which can see in the Status Register, can activate this interrupt line :

- Data available
- End Of Event flag

6. Event Fragment (ROD-ROB link)

6.1 Event Fragment Description

The Read Out Controller has to combine the event fragment for the four PU boards and build complete events to be sent to the ROB. According to the event format specified for the ATLAS DAQ/EF the event fragment format for the ROD would look like this:

Table 6.1—1 : Event Fragment

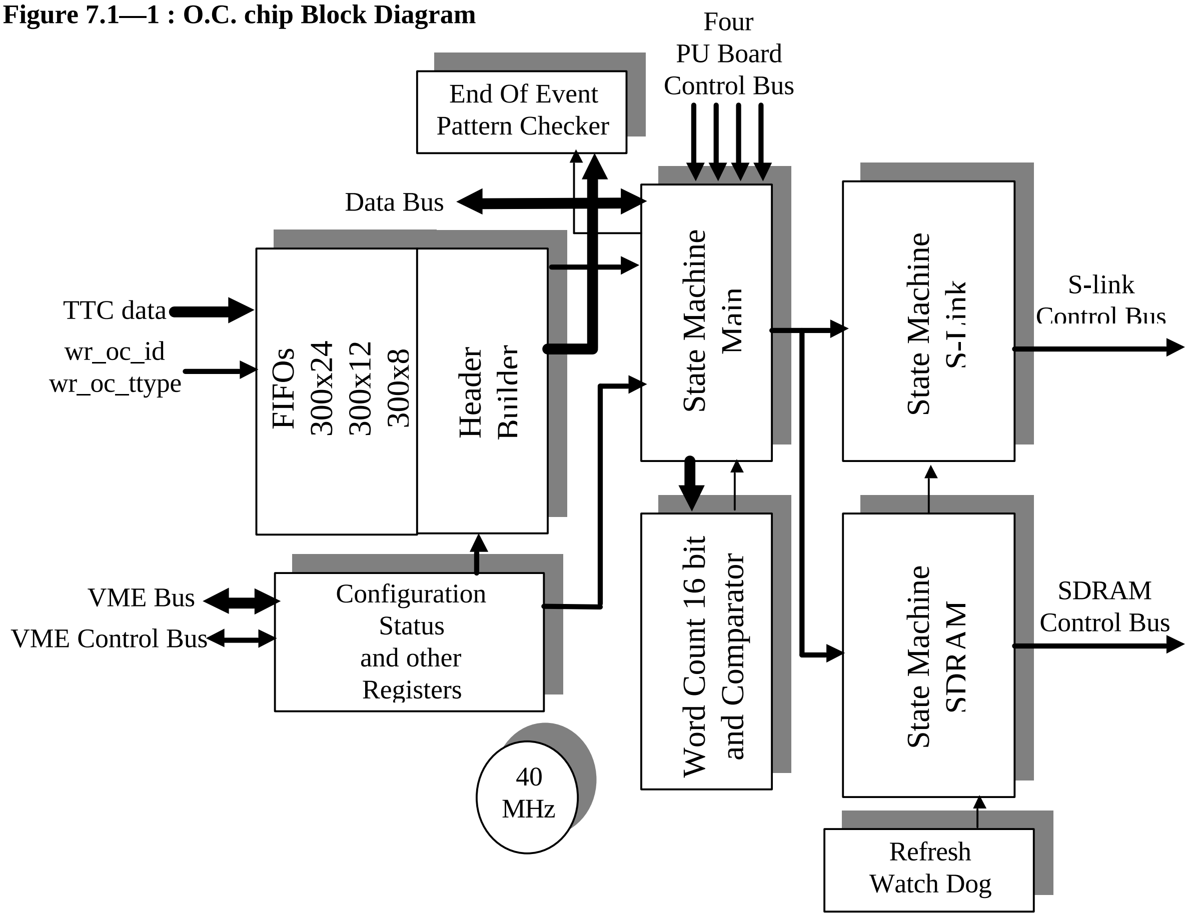
<table>
<tr><td></td><td></td><td></td><td colspan="3"><b>Event Fragment (ROD-ROB link)</b></td></tr>
<tr><td></td><td></td><td>32</td><td colspan="2">31</td><td>0</td></tr>
<tr><td></td><td></td><td>1</td><td colspan="3">Beginning of fragment (S-link frame control word) 0xB0F00000</td></tr>
<tr><td colspan="2" rowspan="8">Header</td><td>0</td><td colspan="3">Start of header marker</td></tr>
<tr><td>0</td><td colspan="3">Header size</td></tr>
<tr><td>0</td><td colspan="3">Format version number</td></tr>
<tr><td>0</td><td colspan="3">Source identifier</td></tr>
<tr><td>0</td><td colspan="3">Level 1 ID</td></tr>
<tr><td>0</td><td colspan="3">Bunch crossing ID</td></tr>
<tr><td>0</td><td colspan="3">Level 1 Trigger Type</td></tr>
<tr><td>0</td><td colspan="3">Detector Event Type</td></tr>
<tr><td colspan="2"></td><td>0</td><td>nb of DSP</td><td>23 PU Mask</td><td>15 Detector Format Version nb. 0</td></tr>
<tr><td rowspan="7">PU n° 1</td><td rowspan="3">Block #0</td><td>0</td><td colspan="3">Word 1 Block #0</td></tr>
<tr><td>…</td><td colspan="3">…</td></tr>
<tr><td>0</td><td colspan="3">Word n Block #0</td></tr>
<tr><td>…</td><td>…</td><td colspan="3">…</td></tr>
<tr><td rowspan="3">Block #5</td><td>0</td><td colspan="3">Word 1 Block #5</td></tr>
<tr><td>…</td><td colspan="3">…</td></tr>
<tr><td>0</td><td colspan="3">Word n Block #5</td></tr>
<tr><td rowspan="7">PU n° 2</td><td rowspan="3">Block #0</td><td>0</td><td colspan="3">Word 1 Block #0</td></tr>
<tr><td>…</td><td colspan="3">…</td></tr>
<tr><td>0</td><td colspan="3">Word n Block #0</td></tr>
<tr><td>…</td><td>…</td><td colspan="3">…</td></tr>
<tr><td rowspan="3">Block #5</td><td>0</td><td colspan="3">Word 1 Block #5</td></tr>
<tr><td>…</td><td colspan="3">…</td></tr>
<tr><td>0</td><td colspan="3">Word n Block #5</td></tr>
<tr><td rowspan="7">PU n° 3</td><td rowspan="3">Block #0</td><td>0</td><td colspan="3">Word 1 Block #0</td></tr>
<tr><td>…</td><td colspan="3">…</td></tr>
<tr><td>0</td><td colspan="3">Word n Block #0</td></tr>
<tr><td>…</td><td>…</td><td colspan="3">…</td></tr>
<tr><td rowspan="3">Block #5</td><td>0</td><td colspan="3">Word 1 Block #5</td></tr>
<tr><td>…</td><td colspan="3">…</td></tr>
<tr><td>0</td><td colspan="3">Word n Block #5</td></tr>
<tr><td rowspan="7">PU n° 4</td><td rowspan="3">Block #0</td><td>0</td><td colspan="3">Word 1 Block #0</td></tr>
<tr><td>…</td><td colspan="3">…</td></tr>
<tr><td>0</td><td colspan="3">Word n Block #0</td></tr>
<tr><td>…</td><td>…</td><td colspan="3">…</td></tr>
<tr><td rowspan="3">Block #5</td><td>0</td><td colspan="3">Word 1 Block #5</td></tr>
<tr><td>…</td><td colspan="3">…</td></tr>
<tr><td>0</td><td colspan="3">Word n Block #5</td></tr>
<tr><td colspan="2"></td><td>0</td><td colspan="3">Status flags from Output Controller (End of Event)</td></tr>
<tr><td colspan="2" rowspan="3">Trailer</td><td>0</td><td colspan="3">Number of Status Elements</td></tr>
<tr><td>0</td><td colspan="3">Number of Data Elements</td></tr>
<tr><td>0</td><td colspan="3">Status Block Position</td></tr>
<tr><td colspan="2"></td><td>1</td><td colspan="3">End of fragment (S-link frame control word) 0xE0F00000</td></tr>
</table>

7. The O.C. chip

7.1 Description and Block Diagram

The Figure 7.1—1 shows the block diagram, which is, built in a FPGA Altera FLEX 10KE.
At the power-up the Main State Machine (MainSM) ask to its two State Machines, State Machine S-Link (SlinkSM) and State Machine SDRAM (SdramSM), under its control to start the boot sequence.
When this boot sequence is finished The MainSM wait that the event builder is ready and that each PU not masked has an event ready.
The TTC data is stored in three 300 words deep FIFO. When there is no FIFO empty the Header Builder start to collect all the words it needs to build the Header. When the header is ready it notify the MainSM.
When all the activated PU and the Header Builder are ready, the data transfer begins through the S-Link or the SDRAM or both, according to the choice made in the Configuration Register.
The header is sent out first followed by one or more PU and last the Trailer.
The first word of the event, which comes from a PU, contains the size of the event. This size is loaded in the comparator, and when the word count reach this size the MainSM is notified that it have to skip to the next PU or finish the transfer by sending the Trailer.
At the end of each PU transfer the End Of Event Pattern Checker is enabled and it loads the pattern which is on the data bus. It checks if this pattern is 0x00000e0e. If not a bit in the Status Register is set. This bit can be reset through the Reset Register.
When all the activated PU are sent its event. The MainSM send the Status Flag and the Trailer which contains the number of Status Elements, the number of Data Elements, and the Status Block Position.
After that it is ready to restart with a new event.

Figure 7.1—1 : O.C. chip Block Diagram



7.2 Pins Description

Table 7.2—1 : Pins Description

Pin Name	Type	Description
addr <0:11>	O	These lines are the Row address (addr<11:0>) or the Column address (addr<9:0>) of the SDRAM, depending of the command.
burst	I	When is high this line indicates that there is a VME access in sequential mode.
cas_n	O	One of the four SDRAM's command lines.
ck	I	General system clock. 40Mhz.
ck_delayed	I	General system clock. 40Mhz. Delayed of about 10ns.
cke	O	This line activates (high) or deactivates (low) the clock of the SDRAM.
count_bank <0:1>	O	Bank Address of the SDRAM.
cs_n	O	One of the four SDRAM's command lines.
data_bus <0:31>	I/O	The main data bus. It allows the chip to receive data from the four PU and it allows the chip to send data to the SDRAM or to the S-link.
dqm <0:1>	O	Used by the SDRAM. These two lines are always driven low, since the SDRAM is a 16 bit version.
en_sdram_b	O	When low this line selects, through an external tri state buffer, the SDRAM as the output buffer.
en_slink_b	O	When low this line selects, through an external tri state buffer, the S-Link as the output buffer.
eoe_flag	O	When high this lines indicates that the "End Of Event" error is occurred. This error arrives when the last word of the event is not 0x00000e0e
evnt_sent	O	A 50ns pulse is sent on this line each time an event is wholly transferred to the output buffer, which can be the SDRAM or the S-link.
fifo_evt_end	O	A 25ns pulse is sent on this line at the end of the event. This line is used by the four PU to decrement their event counter.
fifo_evt_rdy <1:4>	I	When are high this lines indicate that the related PU is ready to send data.
fifo_oen <1:4>	O	When low this line enables the buffer of the related PU.
fifo_rdenn <1:4>	O	When low this line enables the PU's fifo.
ldown	I	After the reset, if is low, this lines indicates that the S-Link is not operational.
lff	I	Link Full Flag. When high this line indicates that the link is full.
link_fail	O	When is high, this lines indicates that the S-Link fails.
lrl <0:3>	I	These S-Link lines are not functional.
oc_conf_wr	I	When is high this line allows to write the Configuration Register through the vme_bus<0:31> lines.
oc_conf_rd	I	When is high this line allows to read the Configuration Register on the vme_bus<0:31> lines.
oc_det_frmtv	I	When is high this line allows to write the Detector Format Version Number Register through the vme_bus<0:31> lines.
oc_evt_typ	I	When is high this line allows to write the Detector Event Type Register through the vme_bus<0:31> lines.
oc_frmtv	I	When is high this line allows to write the Format Version Number Register through the vme_bus<0:31> lines.
oc_interrupt	O	Active high interrupt line.
oc_ram_rd	I	When is high, this line allows to read the SDRAM on the vme_bus<0:31> lines.
oc_ram_rdy	O	When is high, this line indicates that the SDRAM data is extracted and can be read.
oc_reset_reg	I	When is high this line allows to write the Reset Register through the vme_bus<0:31> lines.

oc_src_idt	I	When is high this line allows to write the Source Identifier Register through the vme_bus<0:31> lines.
oc_stat_rd	I	When is high this line allows to read the Status Register through the vme_bus<0:31> lines.
ras_n	O	One of the four SDRAM’s command lines.
rst_b	I	When is low, the chip is reset. All the registers must be reloaded.
ttc_data <0:7>	I	8-bit data bus which allows to write in the input FIFO the TTC information.
uctrl	O	When is low, this lines indicates that the data, which is transmitted to the S-Link, is a control word.
udw <0:1>	O	These two lines are used by the S-Link and defines the data bus width. In our case they are always low to define 32-bit data width.
ureset	O	When is low, this line initiates the S-Link’s reset cycle.
utest	O	When is low, this line switches the S-Link’s LSC (Link Source Card) to test mode.
uwen	O	When is low, this line enables data to be transferred to the S-Link.
vme_bus <0:31>	I/O	32-bit data bus which is connected to the VME interface.
vmerd_oe_b	O	When is low, this line allows the VME external buffer to be activated.
we_n	O	One of the four SDRAM’s command lines.
wr_oc_id	I	When is high, this line allows to write in a FIFO, the BCID (12bits) and the EvtID (24bits), through the ttc_data<0:7> lines.
wr_oc_ttype	I	When is high, this line allows to write in a FIFO, the TTYPE (8bits), through the ttc_data<0:7> lines.

7.3 Pin-Out, alphabetical and numerical order

Table 7.3—1 : Pin-Out

Pin Name	Pin N°		Pin N°	Pin Name
addr0	46		6	oc_reset_reg
addr1	45		7	fifo_rdenn2
addr10	48		8	udw1
addr11	128		9	utest
addr2	44		11	udw0
addr3	43		14	uctrl
addr4	136		17	oc_det_frmtv
addr5	134		18	fifo_rdenn3
addr6	133		19	oc_conf_wr
addr7	132		20	fifo_rdenn4
addr8	131		28	fifo_evt_rdy2
addr9	129		29	fifo_evt_rdy4
burst	196		30	fifo_evt_rdy1
cas_n	54		31	fifo_evt_rdy3
ck	91		33	oc_frmtv
ck_delayed	211		34	lrl2
cke	127		35	lrl3
count_bank0	50		36	lrl0
count_bank1	49		38	lrl1
cs_n	51		39	ureset
dqm0	56		40	lff
dqm1	126		41	ldown

en_sdram_b	80		43	addr3
en_slink_b	215		44	addr2
eoe_flag	218		45	addr1
evnt_sent	214		46	addr0
fifo_evt_end	207		48	addr10
fifo_evt_rdy1	30		49	count_bank1
fifo_evt_rdy2	28		50	count_bank0
fifo_evt_rdy3	31		51	cs_n
fifo_evt_rdy4	29		53	ras_n
fifo_oen1	220		54	cas_n
fifo_oen2	221		55	we_n
fifo_oen3	81		56	dqm0
fifo_oen4	222		61	vme_bus0
fifo_rdenn1	101		62	vme_bus1
fifo_rdenn2	7		63	vme_bus2
fifo_rdenn3	18		64	vme_bus3
fifo_rdenn4	20		65	vme_bus4
ldown	41		66	vme_bus5
lff	40		67	vme_bus6
link_fail	217		68	vme_bus7
lrl0	36		70	vme_bus8
lrl1	38		71	vme_bus9
lrl2	34		72	vme_bus10
lrl3	35		74	vme_bus11
oc_conf_rd	90		75	vme_bus12
oc_conf_wr	19		76	vme_bus13
oc_det_frmtv	17		78	vme_bus14
oc_evt_typ	193		79	vme_bus15
oc_frmtv	33		80	en_sdram_b
oc_interrupt	204		81	fifo_oen3
oc_ram_rd	210		83	oc_src_idt
oc_ram_rdy	84		84	oc_ram_rdy
oc_reset_reg	6		90	oc_conf_rd
oc_src_idt	83		91	ck
oc_stat_rd	212		92	rst_b
ras_n	53		101	fifo_rdenn1
rst_b	92		103	vme_bus16
ttc_data0	188		105	vme_bus17
ttc_data1	187		106	vme_bus18
ttc_data2	186		107	vme_bus19
ttc_data3	185		108	vme_bus20
ttc_data4	184		109	vme_bus21
ttc_data5	183		110	vme_bus22
ttc_data6	182		111	vme_bus23
ttc_data7	181		113	vme_bus24
uctrl	14		114	vme_bus25
udw0	11		115	vme_bus26
udw1	8		116	vme_bus27
ureset	39		117	vme_bus28
utest	9		118	vme_bus29

uwen	223		119	vme_bus30
vme_bus0	61		120	vme_bus31
vme_bus0	175		126	dqm1
vme_bus1	62		127	cke
vme_bus1	174		128	addr11
vme_bus10	72		129	addr9
vme_bus10	163		131	addr8
vme_bus11	74		132	addr7
vme_bus11	162		133	addr6
vme_bus12	75		134	addr5
vme_bus12	161		136	addr4
vme_bus13	76		137	vme_bus31
vme_bus13	159		138	vme_bus30
vme_bus14	78		139	vme_bus29
vme_bus14	158		141	vme_bus28
vme_bus15	79		142	vme_bus27
vme_bus15	157		143	vme_bus26
vme_bus16	103		144	vme_bus25
vme_bus16	156		146	vme_bus24
vme_bus17	105		147	vme_bus23
vme_bus17	154		148	vme_bus22
vme_bus18	106		149	vme_bus21
vme_bus18	153		151	vme_bus20
vme_bus19	107		152	vme_bus19
vme_bus19	152		153	vme_bus18
vme_bus2	63		154	vme_bus17
vme_bus2	173		156	vme_bus16
vme_bus20	108		157	vme_bus15
vme_bus20	151		158	vme_bus14
vme_bus21	109		159	vme_bus13
vme_bus21	149		161	vme_bus12
vme_bus22	110		162	vme_bus11
vme_bus22	148		163	vme_bus10
vme_bus23	111		164	vme_bus9
vme_bus23	147		166	vme_bus8
vme_bus24	113		167	vme_bus7
vme_bus24	146		168	vme_bus6
vme_bus25	114		169	vme_bus5
vme_bus25	144		171	vme_bus4
vme_bus26	115		172	vme_bus3
vme_bus26	143		173	vme_bus2
vme_bus27	116		174	vme_bus1
vme_bus27	142		175	vme_bus0
vme_bus28	117		181	ttc_data7
vme_bus28	141		182	ttc_data6
vme_bus29	118		183	ttc_data5
vme_bus29	139		184	ttc_data4
vme_bus3	64		185	ttc_data3
vme_bus3	172		186	ttc_data2
vme_bus30	119		187	ttc_data1

vme_bus30	138		188	ttc_data0
vme_bus31	120		190	wr_oc_ttype
vme_bus31	137		191	wr_oc_id
vme_bus4	65		193	oc_evt_typ
vme_bus4	171		196	burst
vme_bus5	66		199	vmerd_oe_b
vme_bus5	169		204	oc_interrupt
vme_bus6	67		207	fifo_evt_end
vme_bus6	168		210	oc_ram_rd
vme_bus7	68		211	ck_delayed
vme_bus7	167		212	oc_stat_rd
vme_bus8	70		214	evnt_sent
vme_bus8	166		215	en_slink_b
vme_bus9	71		217	link_fail
vme_bus9	164		218	eoe_flag
vmerd_oe_b	199		220	fifo_oen1
we_n	55		221	fifo_oen2
wr_oc_id	191		222	fifo_oen4
wr_oc_ttype	190		223	uwen