Associative Memory Board

Contents

1. **AM Board functional description**
2. **GLUE tree**
3. **Clock Distribution**
4. **VME Interface**

This note complements SVT Note n.30 (“Amchip2 preliminary specifications”), n.87 (Glue Chips V1.0), n.83 (“GLUE3=AMSGLUE”) and is written with the goal of collect in one place the description of main parts of the AM board.

AM Board functional description

The Associative Memory Board (AMB) is a bank of content-addressable memory for the purpose of fast pattern recognition. It is a part of the Silicon Vertex Tracker (SVT); a trigger processor dedicated to reconstruction of charged particles trajectories at the Level 2 of the CDF trigger. The Associative Memory system is made of the one control board (the Associative Memory Sequencer, AMS) and two Associative Memory Boards (AMB). Its function is to perform pattern recognition, which is to select for further processing only hit combinations that represent good candidates. This is the most computationally-intensive part of SVT data processing, and it is done by comparing the input data with a stored set of patterns in a completely parallel way, using a dedicated custom VLSI Associative Memory chip (AM chip).

The layout of the AM board (9Ux400 VME) is shown in figure 1. The main parts are:

- *Array of 128 AM chips*. The 128 AM chips cover a very large fraction of the usable component area of the AM board, adding the GLUE logic and clock distribution trees left almost no place for the VME logic and routine. As a consequence the AM board was physically implemented as a single–width 9U PCB with 16 connectors for mezzanine cards (*Amplugs*) that carry eight AM chips, low level GLUE0 chip and PLL based clock fanout ('Roboclocks').
- *Glue tree*. The GLUE chips implement pipeline and fanout functions for the operation of array of AM chips. The AM board has three different types of GLUE chips corresponding to different levels of a tree having the AM chips as leaves (see figure 1). At the bottom of the tree the **Glue0** talks to AM chips - *one* Glue0 chip connects *eight* AM chips. The AM board then contains *four* **Glue1** (each connects *four* Glue0) and *one* **Glue2** that connects the *four* Glue1. Glue2 also incorporates the VME interface for the AM board and talks to the AM bus on connector P3*.
- *Clock Distribution tree.* To simplify the problem of clock distribution, only one clock phase is distributed over the board. This is accomplished by using a phase-locked-loop architecture based on PLL clock buffers from Cypress Semiconductors ("Roboclocks"). It provides GLUE chips clocks that aligned both in phase and in frequency with a reference-input clock. By using the programmable skew mode the output clocks from "Roboclocks" can be skewed (advanced or delayed) in the range of up to ±12ns.
- *VME bus interface.* The VME slave interface core has been implemented in a single Xilinx XC 73144 EPLD chip. It was implemented as two finite state machines and performs the two kinds of VME operations:
  - ✓ *OPC* operations: all Amchip opcodes operations
  - ✓ *Special* operations:
    1. Read register 0
    2. Read prom

* There is *additional* **GLUE3** chip, which is mounted on the AM sequencer board to "glue" together two, AM boards which controlled by a single logic.

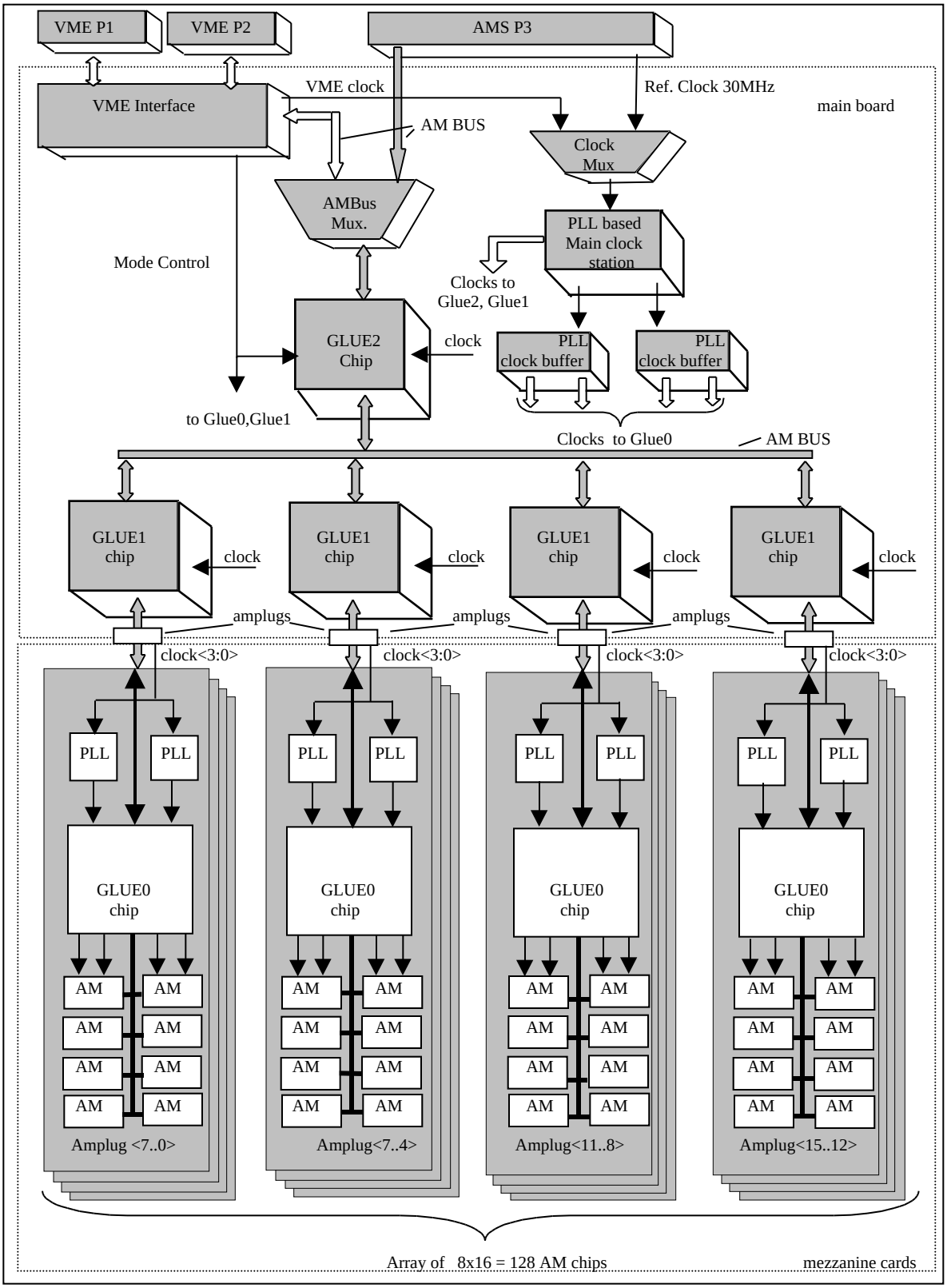


Figure 1. Associative Memory Board block diagram.

3. Switch from P3 mode to VME mode
4. Switch from VME mode to P3 mode
5. Switch from Halt mode to Pipe mode

The AM board can be operated in two different modes:

- 'VME mode' – In this mode the AM board ignores signals coming from the AMS (P3 connector) and operates on an internal VME clock. PLL's clock tree are disabled by using the special TEST feature of the "Roboclocks", that makes them act as simple fixed delays. However the downstream logic of GLUE0 chip is still capable of generating clock phases required for operation AM chips. This mode is used to load the patterns into the associative memory and to test of AM board.
- 'Running mode' or 'PIPE' – The AM board is controlled by the AMS (P3 connector) and runs with the fast clock (30MHz) distributed on the backplane by the AMS. In this mode, the Glue tree distributes the opcodes and the data received on P3 to all Amchips, collects data output by all chips and put them to the P3 bus. This is done through two tree-like structures, an input tree and an output tree; with 128 chips are at the bottom level. During the *Input* phase (see description of Amchip in Appendix A), the AM chips receives an input data/control stream per clock. The content of each word of array AM chips is compared to the content of data bus and, in case of match, the corresponding *hit-bit* is set. In *Count* mode the number of hit-bits set in each pattern is counted. If the number of hit-bits in a pattern is above a given threshold and the hit bit of the plane #5 is set, the *match-bit* is set. The counting is performed in parallel for all the patterns. During the *Output* phase, the addresses of all the patterns having the match-bit set are output, one per clock cycle, in ascending order.

2. GLUE TREE.

Figure 1 shows that the input and output GLUE trees of AM board have three levels Each level consist of a number of identical GLUE chips implemented using Field Programmable Gate Arrays (FPGA) by Quicklogic. Each GLUE chip has ideally two "sides" UP and DOWN, corresponding to the signals going respectively to the higher and lower level in the GLUE tree. At each level there are common input and output busses:

- ⇔ DATA_UP<11:0>, DATA_DW<11:0>
  LAYER_UP<3:0>, LAYER_DW<3:0>,
  OPCODE_UP<3:0>, OPCODE_UP<3:0>,
- ⇔ ADDRESS_UP<(14÷12):0>, ADDRESS_DW<(14÷12):0>

and individual control lines for each chip: DV_UP, DONE_UP, DONE_DW, SEL_UP, DV_DW< 3:0> (DV_DW< 7:0> for GLUE0 chip!), SEL_DW<3:0>(SEL_DW< 7:0> for GLUE0 chip!).

The GLUE0 chips handles 8 AMchips – so, 8 bi-directional Data Buses DB<11:0> of Amchips wired together and connected to the bi-directional Data Bus

DATA_DW<11:0> of GLUE0. The control lines SEL_DW<7:0> are used to enable a given Amchip by prioritizing signals from lines DV_DW< 7:0> (when OPcode = OUTPUT) or by decoding of the GLUE0 ADDRESS_UP bus 3 most significant bits (when Opcode ≠OUTPUT). Lines DV_DW< 7:0> connected to the pins OR of Amchips.

The GLUE1 chips handles 4 GLUE0 chips – therefore, 4 bi-directional Data Buses DATA_UP<11:0> of GLUE0 wired together and connected to the bi-directional Data Bus DATA_DW<11:0> of GLUE1. The four control lines SEL_DW<3:0> are used to enable a given CLUE0 chip by prioritizing signals from lines DV_DW< 3:0> (when OPcode = OUTPUT) or by decoding of the GLUE1 ADDRESS_UP bus 2 most significant bits (when Opcode ≠OUTPUT). Lines DV_DW< 3:0> are connected to the pins DV_UP of GLUE0.

Accordingly, the GLUE2 chips handles 4 GLUE1 chips – therefore, 4 bi-directional Data Buses DATA_UP<11:0> of GLUE1 wired together and connected to the bi-directional Data Bus DATA_DW<11:0> of GLUE2. The four control lines SEL_DW<3:0> are used to enable a given CLUE1 chip by prioritizing signals from lines DV_DW< 3:0> (when OPcode = OUTPUT) or by decoding of the GLUE2 ADDRESS_UP bus 2 most significant bits (when Opcode ≠OUTPUT). Lines DV_DW< 3:0> connected to the pins DV_UP of GLUE1.

The working modes of GLUE chips controlled by flags generated by the VME interface - MODE[1:0]:

- ⇔ Mode PIPE MODE[1:0] = 0x0
- ⇔ READ MODE[1:0] = 0x1
- ⇔ WRITE MODE[1:0] = 0x2
- ⇔ HALT MODE[1:0] = 0x3

Two finite state machines inside of each GLUE chip, one for the down and one for the up side handle data moving up the tree. All FSM's work on the same common clock sent by the AM sequencer (P3 connector). The down FSM of the GLUE0 also controls the clock lines to the Amchips.

For Mode PIPE there are two different main tasks and two accordingly different protocols for communication of GLUE chip:

- ✓ Latches OPCODE bus, LAYER bus and DATA bus from UPPER side in synchronous register and send them to DOWN side *during non OUTPUT mode*.

  To ensure fast and reliable operation, the input (downward) tree operates as a simple pipeline synchronized on the board clock, moving the data one level down at every clock cycle without handshake.
- ✓ Transfer the AM addresses (roads) from DOWN side to UPPER side *during OUTPUT mode*.

  The output (upward) pipeline works instead with synchronous handshake protocol (DONE_/DV_/SEL_ protocol) and allows sequential queueing of data which may be present simultaneously on several tree nodes at the same level, one word is pushed up the tree each 3 clock cycles.

The reason for two different protocols is twofold: on one side moving data up the tree from the 128 AM chips to the single AAM sequencer requires additional time to switch the common address bus from one chip to the other. On the other the typical

data flow in the system is of several hundreds of data words going in and very few coming out (many hits, few track). Therefore the GLUE chip is designed to keep the clock cycle as short as possible (in each clock cycle signals only go from one driving to one receiving chip) allocating more cycles for the more complex output function.

3. Clock Distribution

Clock is distributed through an independent tree. A two-phase clock that called CLKA and CLKB drives the Amchip. Their relative timing are identified by for distinct phases t_0, t_1, t_2, t_3 in each clock cycle CLK t_{clk1} (see Appendix A). Moreover, for a proper handshake of the data exchange between the GLUE0 chip and Amchips an additional clock phase t_{clk3} need to be delivered to the GLUE0 chip. This is necessary because during the INPUT phase DATA_DW must be valid at the Amchip on the falling edge of CLKB. Both clock distribution and multiple phases generation are accomplished by using PLL based clock buffers from Cypress Semiconductors ('Roboclocks'). For full speed performance of AMchips, the clock edges must be placed with an accuracy of couple of nanosecond and so everywhere on the board. To simplify the problem of clock distribution, only one clock phase is distributed over the board (master clock), and the 5 remaining phases are locally generated on where needed with definite relative timing.

The main parts of phase generation and clock generation subsystem are (see Figure 2):

- **Clock Switch Multiplexer** *U13* (74FCT257). During **PIPE mode** under Associative Memory Sequencer (AMS) control the timing of all operations based on one 3*0MHz Base Clock* from P3 connector. During **VME mode** the clock from P3 connector is switched off and the clock to the Master Clock Station is run directly under VME control.
- **Master Clock Station** based on Programmable Skew PLL Clock Driver ('Roboclock') *U38* (CY7B992) and two Low Skew Clock Buffer *U28,U29* (CY7B9920). The timing of each of the 3 phases for GLUE2 chip (GL2CLK2), GLUE1 chip (GL1CLK1), GLUE0 chip (GL0CLK<i>) and skew for all of them with respect to the input Base Clock (FB) can be adjusted by controlling the individual jumper-programmable delays of each Roboclock output, in steps of about 1ns. Table 1 shows the possible delays for each outputs of 'Roboclock' as determinated by the status of jumper J2÷**J9.** Jumper **J2÷J9** are located on Amboard near front panel (upper corner).

Jumpers		Name of clock skew	Skew as function of jumpers Set								
			L L	L M	L H	M L	M M	M H	H L	H M	H H
J2	J3	GL1CLK1 to BASE CLOCK	-4t	-3t	-2t	-1t	0t	+1t	+2t	+3t	+4t
J4	J5	GL2CLK1 to BASE CLOCK	-4t	-3t	-2t	-1t	0t	+1t	+2t	+3t	+4t
J6	J7	ALL to BASE CLOCK	:2	-6t	-4t	-2t	0t	+2t	+4t	+6t	:4
J8	J9	GL2CLK0 to BASE CLOCK	:2	-6t	-4t	-2t	0t	+2t	+4t	+6t	:4

Table 1. Clock skew with respect to the input Base Clock as function of jumpers Set. L - indicates a connection to GND (jumper set to left position); H- indicates a connection toVcc=+5V (jumper set to right position); M - indicates an

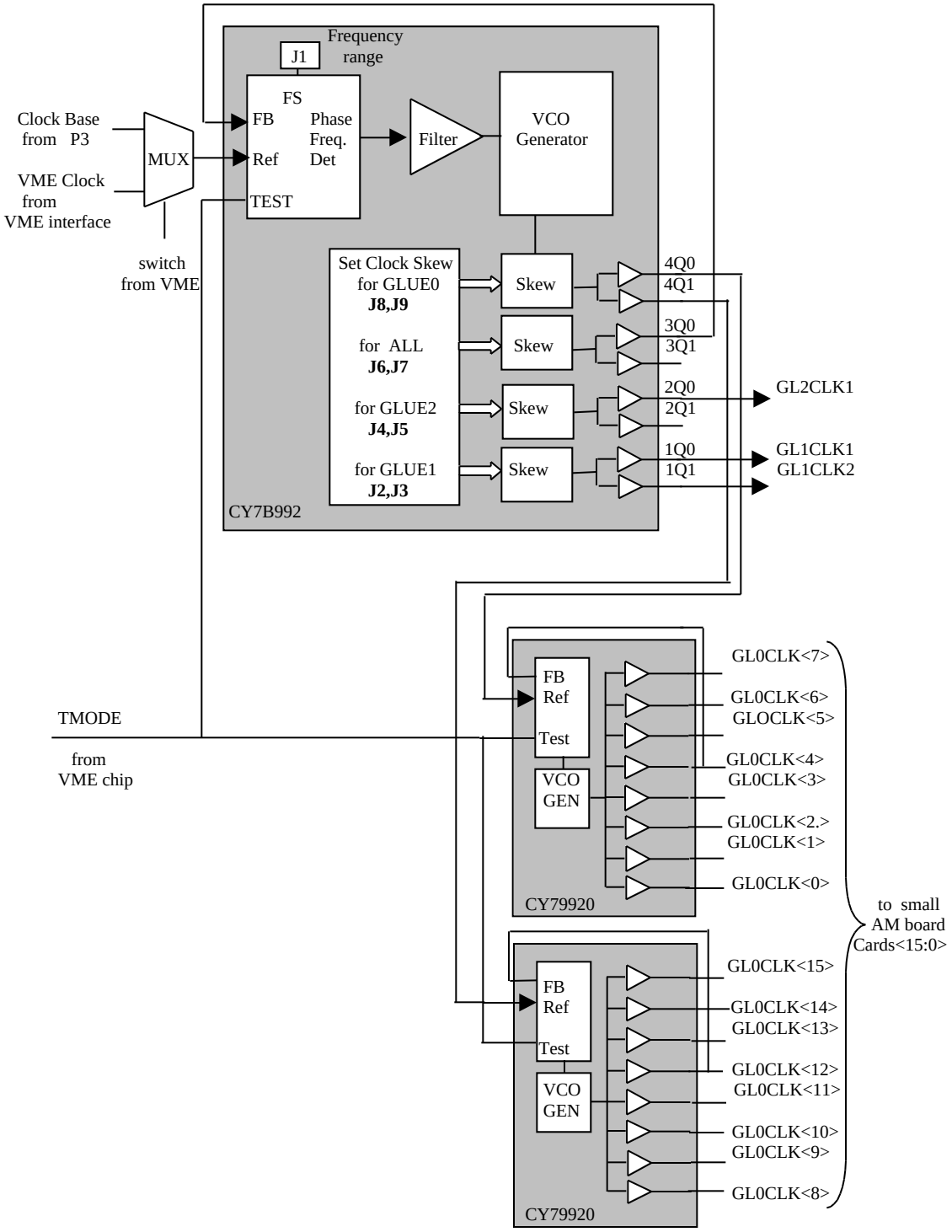


Figure 2. AM board Clock Distribution Tree. Master Clock Station.

Figure.3

open connection. Upper string of jumper set in Table is corresponding to J3,J5,J7,J9 and bottom is corresponding to J2,J4,J6,J8.

- **AMplug clock scheme** are (see Figure 3) based on the two Roboclock *U2,U3* (CY7B992) and additional internal logic GLUE0 chip *U4* (CY7C3386A). Each Roboclock produces eight outputs with 4 adjustable phase shifts with respect to the input clock, also allowing zero or negative delays. Each of the eight phases from the two Roboclocks, independently controlled by **J11÷J26** jumpers, is used to fix the edge position of the AM chip CLKA,CLKB signals and for GLUE0 pipeline synchronization CLK1,CLK3. The **J11÷J26** jumpers are common for all AMplug. Tables 2,3 shows the possible delays for each outputs of 'Roboclock' as determinated by the status of jumper **J11÷J26.** Jumper **J11÷J18** are located on the middle of Amboard near front panel and **J19÷J26** are located on bottom corner of Amboard near front panel too.

Jumpers		Name of clock skew with respect to the input clock	Skew as function of jumpers Set								
			L L	L M	L H	M L	M M	M H	H L	H M	H H
J11	J12	Falling edge of CLKA	-4t	-3t	-2t	-1t	0t	+1t	+2t	+3t	+4t
J13	J14	CLK1	-4t	-3t	-2t	-1t	0t	+1t	+2t	+3t	+4t
J15	J16	ALL	:2	-6t	-4t	-2t	0t	+2t	+4t	+6t	:4
J17	J18	Rising edge of CLKA	:2	-6t	-4t	-2t	0t	+2t	+4t	+6t	:4

Table 2. Clock skew with respect to the input Base Clock as function of jumpers Set. L - indicates a connection to GND (jumper set to left position); H- indicates a connection toVcc=+5V (jumper set to right position); M - indicates an open connection. Upper string of jumper set in Table is corresponding to J12, J14, J16, J18 and bottom is corresponding to J11, J13, J15, J17.

Jumpers		Name of clock skew with respect to the input clock	Skew as function of jumpers Set								
			L L	L M	L H	M L	M M	M H	H L	H M	H H
J19	J20	Rising edge of CLKB	-4t	-3t	-2t	-1t	0t	+1t	+2t	+3t	+4t
J21	J22	CLK3	-4t	-3t	-2t	-1t	0t	+1t	+2t	+3t	+4t
J23	J24	ALL	:2	-6t	-4t	-2t	0t	+2t	+4t	+6t	:4
J25	J26	Falling edge of CLKB	:2	-6t	-4t	-2t	0t	+2t	+4t	+6t	:4

Table 3. Clock skew with respect to the input Base Clock as function of jumpers Set. L - indicates a connection to GND (jumper set to left position); H- indicates a connection toVcc=+5V (jumper set to right position); M - indicates an open connection. Upper string of jumper set in Table is corresponding to J20, J22, J24, J26 and bottom is corresponding to J19, J21, J23, J25.

During **VME mode** the PLL's are disabled using the TEST feature of the Roboclock, that makes them act as simple fixed delays. But AMplug clock logic however is still capable of generating 4 phases for AM chips. When the board is switched to P3 mode,

it is held in a wait state for about a millisecond, with all clocks to the AM chips disabled, to give time to all PLL's to lock-in (see VME State Machine Diagram).

4. VME Interface

The VME slave interface core has been implemented in a single Xilinx XC 73144 EPLD chip. It is organized using two FSM, one called *common* and the other *custom*. The *common* FSM recognize that the master is addressing the board and the type of data transfer (Single Word (SW), Block Transfer (BLT) …) and performs all the standard operations required from the VME protocol. (see figure 4). The common FSM could be used without any change for other VME slave interfaces, while the specific operations to be performed by the board are defined in the *custom* FSM.

The two FSM interact together via the GO-OK protocol: in state S4 of *common* FSM the GO signal becomes true and this is the start signal for the custom FSM. When custom FSM ends its job returns OK signal to the common FSM, which moves from S4 and ends the cycle.

The custom FSM is shown in figure 5. There're 7 branches of states departing from STATE1, named from A to G. VME operations can be grouped as OPC operations and SPECIAL operations. OPC operations are related to operational codes for Amchip: there is one possible operation for each opcode. In the picture these operation are performed by the states in branch A for what concerns all opcode excluding output. Opcode output operation is executed from the states in branch F, however these is only present for completeness, because opcode output operations are not working in VME mode, due to glue chips architecture.

SPECIAL operations include all VME operations not related to opcodes. These operations are:

- Prom reading (see branch B)
- P3 vs VME switch (see branch C)
- VME vs P3 switch (see branch D)
- Mode Halt to mode Pipe (see branch E)
- Reg0 read (see branch G)

VME Addresses map, valid operations and related addresses see in Appendix C:

Appendix A.

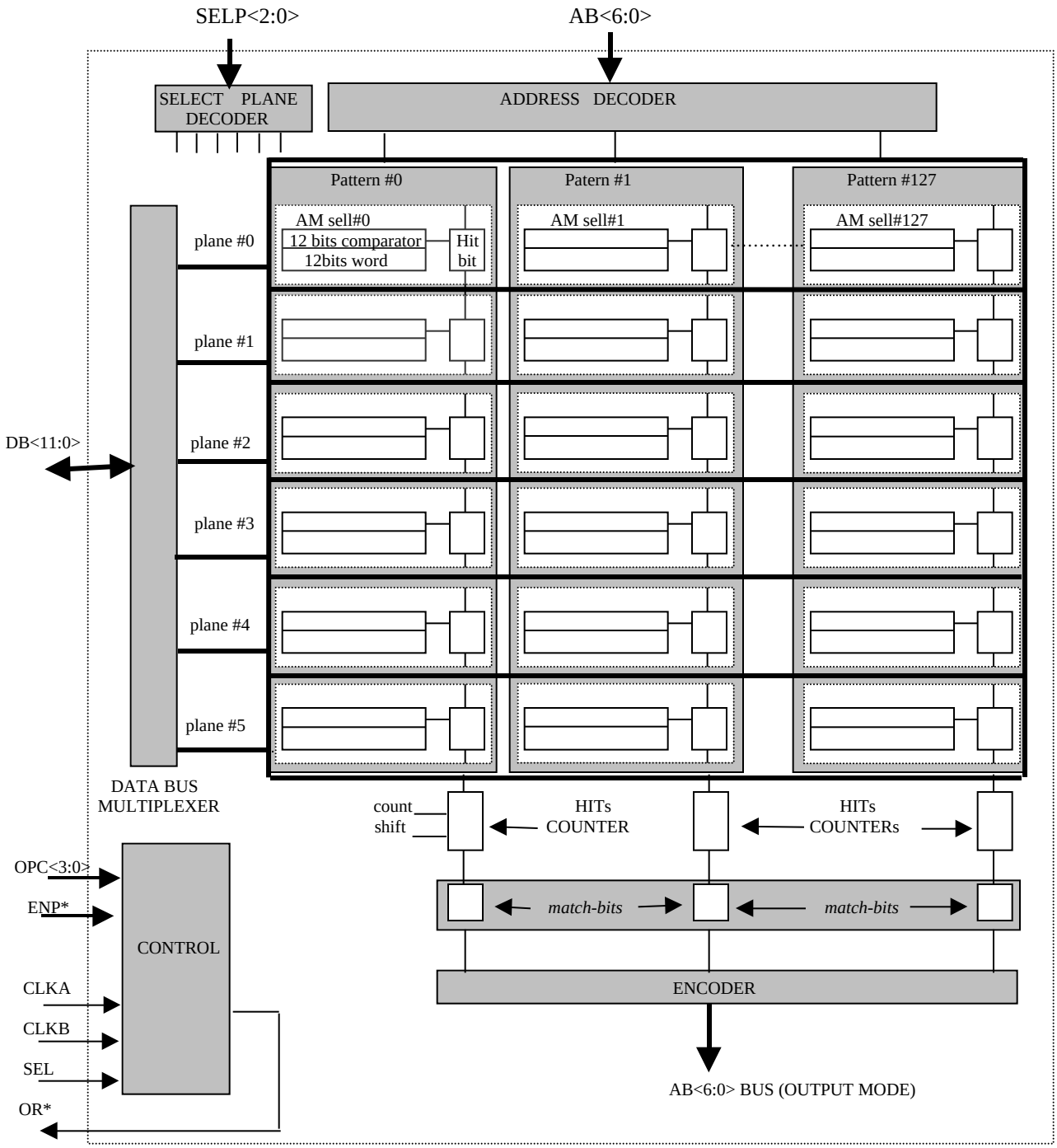
AMchip2 SPECIFICATIONS

Fabio Morsani, Diego Passuello, Luciano Ristori

BRIEF FUNCTIONAL DESCRIPTION

The AMchip2 contains an array of static RAM words. Each word is 12 bit wide. The array consists of 6 columns and 128 rows for a total of 768 words. Each row is called a *pattern* (#0 to #127) and each column is called a *plane* (#0 to #5). The data bus is 12 bit wide so that one pattern is read or written in six clock cycles. Each RAM word is equipped with a comparator and a flip-flop called *hit-bit*. There are six hit-bits per pattern: they form the *hit-register*. There is one hit-register per pattern.

While functioning in *input* mode, the content of each word is compared to the content of the data bus and, in case of match, the corresponding hit-bit is set. In *count* mode the number of hit-bits set in each pattern is counted. If the number of hit-bits in a pattern is above a given threshold **and** the hit bit of the plane #5 is set, the *match-bit* is set. If the hit bit of the plane #5 is not set, the counting result will be equal to 0 for the relative patterns. There is one match-bit per pattern. The counting is performed in parallel for all the patterns. The plane #5 **and** feature can be enabled or disabled: in the latter case the plane #5 is handled just like the other ones. During the *output* phase, the addresses of all the patterns having the match-bit set are output, one per clock cycle, in ascending order. During the *input* phase, the AMchip2 receives an input data/control stream per clock cycle consisting of 12 bits for the data bus, 3 bits for the plane selection, 1 bit to enable or disable the comparision. This generates an output data stream during the following output phase.



Appendix A Figure 1. AM chip internal Structure.

I/O PINS

DB<11:0> : the **D**ata **B**us, active high, multiplexed. It is used for data transmission in *Read* and *Write* operations performed during the *initialization* phase when the patterns are loaded into the memory. It is used in the *input* phase to input data to be compared to all patterns stored in memory. The Data Bus is internally multiplexed/demultiplexed, in order to be connected to a given plane, according to the *SELP<2:0>* bits. The Data Bus is always in input mode when *CLKA* is high (see TIMING section).

SELP<2:0>: the **SEL**ected **P**LANE bits, active high, latched at the falling edge of *CLKA*. They are used to select a plane during the *Read*, *Write* and *Input* cycles. Planes from #0 to #5 can be selected. Attempt to select non-existing planes (#6 and #7) will leave floating internal data bus during *Read* cycles (to be avoided, if possible).

ENP*: the **EN**able **P**lane comparision bit, active low, latched at the falling edge of *CLKA*. It is used during the *Input* phase. Enables comparison between the data present on *DB<11:0>* and the data stored into the selected plane. If the *ENP** bit is inactive for a given *Input* cycle, hit-bits of the selected plane are not set irrespective of *DB<11:0>* contents.

AB<6:0>: the **A**ddress **B**us, active high. Each pattern is identified by a 7 bit address. There are 2^7=128 patterns in one chip. The address bus is used as input during *Read* and*Write* operations and as output during the *Output* operation (see the following section). The Address Bus is always in input mode when *CLKA* is high (see TIMING section).

OPC<3:0>: the **OPC**ode, active high, latched at the falling edge of *CLKA*. Specifies the particular operation to be performed during the current clock cycle as specified in table 1. Details on each operation are given in the following section.

CLKA, CLKB: the two-phase clock, active high. The chip is driven by a two-phase clock. All operations are synchronized to these clock signals as specified further. The exact timing of these signals will be adjusted to optimize device performance. The static circuitry of AMchip2 allows to slow down to 0 the clock frequency. Hovewer this is not possible during Input cycles, between the falling edge of CLKA and the rising edge of CLKB.

OR*, SEL*: respectively **O**utput **R**eady and chip **SEL**ect, active low. These signals are used in the *Output* phase to synchronize the output data stream and during the read/write operations.

*OR** is asserted by the chip to signal the presence of data to be output. It is removed once all data have been output.

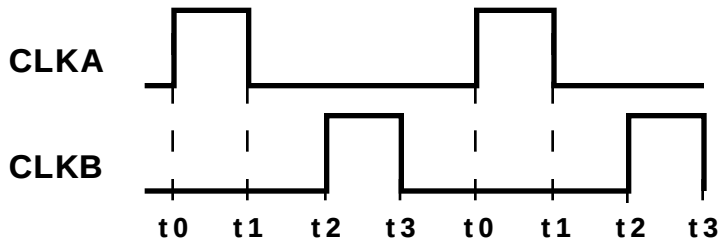
*SEL** is asserted to the chip to enable data output on the Address Bus (see below).

*SEL** is also used during *Read* and*Write* operations (see below).

*SEL** is latched at the falling edge of *CLKA*.

TIMING: general description.

The AMchip is driven by a two-phase clock. The two phases are called *CLKA* and *CLKB* and their relative timing is shown below.



The four transition times are identified as t_0, t_1, t_2 and t_3. In what follows we will use the expression "stable at t_1", "stable during *CLKA*", "stable during *CLKB*" to mean "stable between t_1-Δt_s and t_1+Δt_h", "stable between t_0-Δt_s and t_1+Δt_h", "stable between t_2-Δt_s and t_3+Δt_h", where the Δt_h and the Δt_s depend on the case. The *OPC<3:0>*, the *SEL**, the *ENP** and the *SELP<2:0>* are latched internally on the falling edge of *CLKA* (t_1): the latch is transparent between t_0 and t_1. This signals must therefore be valid and stable at t_1 considering also that the *OPC<3:0>* decoding begins at t_0, requires few nanoseconds to be ended, and during *CLKA* the *Read* and *Input* operations are initialized. To avoid any conflict on the Data and Address Busses, they enter in input mode as soon as possible after t_0, returning in the appropriate mode as soon as possible after t_1.

DESCRIPTION OF OPERATIONS

For all operations, *OPC<3:0>* bits must be stable at t_1.

Read: Reads one word from memory. The plane is selected by *SELP<2:0>*, the pattern address must be specified on the *AB<6:0>*, the data are outputted on the *DB<11:0>* if the *SEL** is asserted (*SEL**=0). See the relative timing diagrams.

Write: Writes one word to memory. The plane is selected by *SELP<2:0>*, the pattern address must be specified on the *AB<6:0>* and the data to be written on the *DB<11:0>*. If the *SEL** is not asserted (*SEL**=1) the *Write* operation has no effect. The *DB<11:0>* content is written continously into the memory cells during *CLKB* so it is necessary to hold the *DB<11:0>* to the desired value a given time before t_3. See the relative timing diagrams.

Input: The data are compared to words stored in the selected (by *SELP<2:0>*) plane if the *ENP** is active. Hit-bits corresponding to matching words are set just after t_2 so *DB<11:0>* bits must be stable at t_2. See the relative timing diagrams.

Enable: enables the special feature of plane #5. With this feature enabled, the *Shift/Count* operations are effective only in the patterns having the plane #5 hit-bit set. AB<6:0> and DB<11:0> are in input mode; SELP<2:0>, ENP*, SEL* have no effect.

Disable: disables the special feature of plane #5. With this feature disabled, the *Shift/Count* operations are effective independently from the plane #5 hit-bit status. AB<6:0> and DB<11:0> are in input mode; SELP<2:0>, ENP*, SEL* have no effect.

Clear Hit-Registers: All hit-bits are cleared. AB<6:0> and DB<11:0> are in input mode; SELP<2:0>, ENP*, SEL* have no effect.

Clear Counter and Load Shift Registers: All the counters (one per pattern) are cleared. The shift registers (one per pattern) are loaded with the Hit-Registers content. AB<6:0> and DB<11:0> are in input mode; SELP<2:0>, ENP*, SEL* have no effect.

Count: Shifts all the Shift Registers and increments the corresponding counters if and only if the bit falling off is set and the hit bit of the plane #5 is set (if such a feature is enabled). Each *Count* operation after a *Clear Counter and Load Shift Registers* looks at hit-bits corresponding to different planes in the sequence #5, #4, .. ,#0. So, it is possible "to skip" a given plane during the *Count* phase using a *Shift* instead of a *Count* (see below). Six clock cycles are needed to count all hit-bits in the six planes. All counters corresponding to all patterns are incremented in parallel. When one counter reaches six, the corresponding match-bit is set.

The *Count* operation destroys the Shift Registers content, filling it with zeros. Anyway, the Shift Registers can be re-loaded with the original content, because the Hit-Registers content is untouched until next *Input* or *Clear Hit-Registers* operations. AB<6:0> and DB<11:0> are in input mode; SELP<2:0>, ENP*, SEL* have no effect. OR* is asserted (=0) if some enabled pattern (read or written after last *Init*) has the match-

bit set. See the relative timing diagrams.

Shift: Shifts all the Shift Registers, as in *Count* cycle, and increments all counters. When used during the *Count* phase at plane #n (n=0,..,5), the *Shift* is equivalent to a *Count* with all hit-bits of the plane #n set (all counters are incremented). This allows "to skip" a given plane forcing it to have always all patterns matching the input data. Six *Count* cycles will set the match-bits of all the patterns with six hit-bits on. A subsequent *Shift* cycle will set the match-bits of all the patterns with five hit-bits. Another *Shift* cycle will set the match-bits of all the patterns with four hit-bits and so on. AB<6:0> and DB<11:0> are in input mode; SELP<2:0>, ENP*, SEL* have no effect. OR* is asserted (=0) if some enabled pattern (read or written after last *Init*) has the match-bit set. See the relative timing diagrams.

Init: Disables all patterns for the *Input/Output* operations. Must be performed once before loading the memory with patterns. Each pattern is enabled by a *Write* or a *Read* to the corresponding address. AB<6:0> and DB<11:0> are in input mode; SELP<2:0>, ENP*, SEL* have no effect. OR* is deasserted (=1). See the relative timing diagrams.

Reset Handshake: Resets the internal sequential logic that controls the output phase causing subsequent *Output* cycles to begin from the first matching pattern. Needed at least once, after the *Init* . AB<6:0> and DB<11:0> are in input mode; SELP<2:0>, ENP*, SEL* have no effect. OR* is asserted (=0) if some enabled pattern (read or written after the last *Init*) has the match-bit set.

Output: Output of data stream on the Address Bus: the available patterns addresses are output on the *AB<6:0>* bus in ascending order. The operations are controlled by two signals: *OR** and *SEL**.

*OR** is asserted as soon as there is at least one match-bit set (during *Count/Shift* or *Reset Handshake* operations). Addresses of all patterns having the match-bit set are output in sequence during subsequent *Output* cycles. The *SEL** signal enables output of the current address and causes the advancement to the next pattern. The next pattern will be output during the following *Output* cycle. If *SEL** is inactive, data are held within the chip and the address bus goes high-z. *OR** is removed as soon as all patterns with match-bits set have been output. See the relative timing diagrams.

If *OR** is inactive, there is no data to be output.

If *OR** is active, there is data to be output and the following happens at every clock cycle:

1. *OPC<3:0>* and must be stable at t_1 and is latched at the falling edge of *CLKA*.

2. if *OPC<3:0>* is different from *Output* then the corresponding operation is performed using the value of *SEL** latched at the falling edge of *CLKA*.

3. if *OPCODE* is equal to *Output* then *SEL** is used to asynchronously enable the driving of the address bus until the following t_0. The delay between *SEL** assertion (deassertion) and the presence (absence) of valid data on the address bus is indicated by T_{SAB}.

4. if *SEL** is active at t_3, the pattern address is internally advanced to the following one having the match-bit set. The new address will be available beginning from t_3+T_{ABchg}. If no more patterns are available, then *OR** is deasserted. The delay between t_3 and the removal of *OR** is indicated by T_{ORoff}.

See the relative timing diagrams.

NOTE: In order to correctly initialize the *OR** a *Clear Counter and Load Shift Registers* operations must be done during the chip initialization phase, before patterns loading. The correct initialization operations to be done are hence, in sequence:

1 *Init*	disable all pattern, *OR**=1 (deasserted)
2 *Reset Handshake*	(*OR** remains deasserted)
3 *Clear Counter and ..*	it clears also the match-bit (*OR** remains deasserted).
4 *Write(s)/Read(s)*	*OR** remains correctly deasserted.

TABLE 1

OPCODE	***Operation***
0	*No Operation (NOP0)*
1	*Read*
2	*Output*
3	*Write*
4	*Input*
5	*Clear Hit-Registers*
6	*No Operation (NOP6)*
7	*Shift*
8	*Count*
9	*Clear Counter and Load Shit Registers*
10	*Init*
11	*Disable*
12	*Reset Handshake*
13	*Enable*
14	*No Operation (NOP14)*
15	*No Operation (NOP15)*

TIMINGS (in parenthesis the #opcode(s) involved).

T_{OPs}	OpCode setup time (all)	OpCode valid & stable to falling edge of CLKA.
T_{OPh}	OpCode hold time (all)	OpCode change to falling edge of CLKA.
T_{SELs}	SEL* setup time (1, 3)	SEL* valid & stable to falling edge of CLKA.
T_{SELh}	SEL* hold time (1, 3)	SEL* change to falling edge of CLKA.
T_{ENPs}	ENP* setup time (4)	ENP* valid & stable to falling edge of CLKA.
T_{ENPh}	ENP* hold time (4)	ENP* change to falling edge of CLKA.
T_{SELPs}	SELP setup time (1, 3, 4)	SELP valid & stable to falling edge of CLKA.
T_{SELPh}	SELP hold time (1, 3, 4)	SELP change to falling edge of CLKA.
T_{DBd}	Data Bus output delay (1)	rising edge of CLKB to output data valid & stable.
T_{DBh}	Data Bus output hold (after a 1)	rising edge of CLKA to Data Bus in Hi-Z state.
T_{DBon}	Data Bus from Hi-Z to Drive (1)	falling edge of CLKA to output data valid & stable.
T_{ABh}	Address Bus hold (after a 2)	rising edge of CLKA to Addr. Bus in Hi-Z state.
T_{ABd}	Address Bus output delay (2)	falling edge of CLKA to output data valid & stable.
T_{ABchg}	Address Bus output change (2)	falling edge of CLKB to output data change.
T_{DBWs}	Data Bus input setup (3)	falling edge of CLKB to input data valid & stable.
T_{DBWh}	Data Bus input hold (3)	falling edge of CLKB to input data change.
T_{DBIs}	Data Bus input setup (4)	input data valid & stable to rising edge of CLKB.
T_{DBIh}	Data Bus input hold (4)	input data change to rising edge of CLKB.
T_{ABRWs}	Address Bus input setup (1,3)	input address valid & stable to rising edge of CLKB.
T_{ABRWh}	Address Bus input hold (1, 3)	input address change to falling edge of CLKB.
T_{SAB}	Address Bus output disable (2)	SEL* deassertion to Addr. Bus in Hi-Z state.
T_{Sunchg}	SEL* deass. for AB unchanged (2)	SEL* deassertion to avoid AB change to falling edge of CLKB.
T_{Schg}	SEL* assertion for AB changed (2)	SEL* assertion to have AB change to rising edge of CLKB.
T_{OVLP}	Overlap between CLKA & CLKB	falling edge of CLKA to rising edge of CLKB.

T_{ORon}	OR* assertion delay (8, 7, 12)	rising edge of CLKB to to OR* assertion.
T_{OROff}	OR* deassertion delay (2)	falling edge of CLKB to OR* deassertion.
T_{ORInit}	OR* deassertion delay (13)	rising edge of CLKB to OR* deassertion.
T_{ORrh}	OR* deassertion delay (12)	rising edge of CLKB to OR* deassertion.

Notes on parameters sign.

Example 1: rising (or falling) edge of CLKA (or CLKB) to *event*

if *event* is measured to occurr before the edge, the sign is **-**

Example 2: *event1* to *event2*

if *event2* is measured to occurr after *event1*, the sign is **+**

Example 3: overlap between CLKA and CLKB

if the overlap exists, the sign is **-**

Measurements on 22 devices @ Vdd = 5 Volts, Temp = 23±2 ^{o}C, RH = 50%. (TO BE UPDATED)

The outputs are loaded whith 5mA sink/source active loads. The clock timing used has (t_1-t_0) = (t_2-t_1) = (t_3-t_2) = (next _t_0-previous_t_3).

Parameter	Mean (ns)	Min (ns)	Max (ns)
T_{OPs}	- 3.405	- 3.512	- 3.300
T_{OPh}	0.026	- 0.013	0.050
T_{SELs}	- 2.853	- 2.938	- 2.738
T_{SELh}	- 0.712	- 0.750	- 0.675
T_{ENPs}	- 0.585	- 0.625	- 0.563
T_{ENPh}	- 0.588	- 0.625	- 0.550
T_{SELPs}	- 0.845	- 0.875	- 0.813
T_{SELPh}	- 0.158	- 0.188	- 0.113
T_{DBd}	13.710	13.450	13.913
T_{DBh}	3.150	3.088	3.187
T_{DBon}	4.360	4.212	4.462
T_{ABh}	3.627	3.512	3.713
T_{ABd}	7.649	7.537	7.737
T_{ABchg}	9.839	9.587	10.125
T_{DBWs}	0.040	-0.050	0.100
T_{DBWh}	0.045	0.000	0.125
T_{DBIs}	3.420	3.275	3.513
T_{DBIh}	4.677	4.500	4.825
T_{ABRWs}	-10.341	-11.138	- 9.937
T_{ABRWh}	- 9.861	-10.025	- 9.675
T_{SAB}	3.027	3.100	2.937
T_{Sunchg}	- 3.324	- 3.400	- 3.225
T_{Schg}	-10.543	-10.825	-10.238
T_{OVLP}	- 2.435	- 2.588	- 2.250
T_{ORon}	12.848	12.525	13.050
T_{ORoff}	7.579	7.350	7.775
T_{ORInit}	12.093	11.825	12.325
T_{ORrh}	10.415	10.225	10.588

Maximum frequencies (MHz).

Remember that the clock timing used has (t_1-t_0) = (t_2-t_1) = (t_3-t_2) = (next _t_0-previous_t_3).

OpCode	Mean	Min	Max
Read+Write	23.17	22.20	24.40
Input+Count+Output	24.91	23.30	27.00
Input		66.70	

Idd (mA).

	STATIC		
OpCode	**Mean**	**Min**	**Max**
Init	0.399	0.392	0.408
Reset Handshake	0.400	0.392	0.410
Clear Counter and ...	0.398	0.391	0.409
Write	0.432	0.421	0.441
Read	0.536	0.454	0.752
Clear Hits	0.399	0.392	0.405
Input plane #0	0.464	0.451	0.471
Input plane #1	0.465	0.458	0.472
Input plane #2	0.466	0.455	0.477
Input plane #3	0.466	0.456	0.473
Input plane #4	0.465	0.455	0.474
Input plane #5	0.465	0.459	0.471
Count	0.432	0.425	0.442
Output	0.466	0.456	0.478
Nop0	0.466	0.455	0.478
Nop6	0.397	0.392	0.407
Nop13	0.364	0.357	0.373
Nop11	0.364	0.356	0.372
Nop14	0.364	0.355	0.374
Nop15	0.330	0.325	0.336
Shift	0.363	0.356	0.366

5 MHz

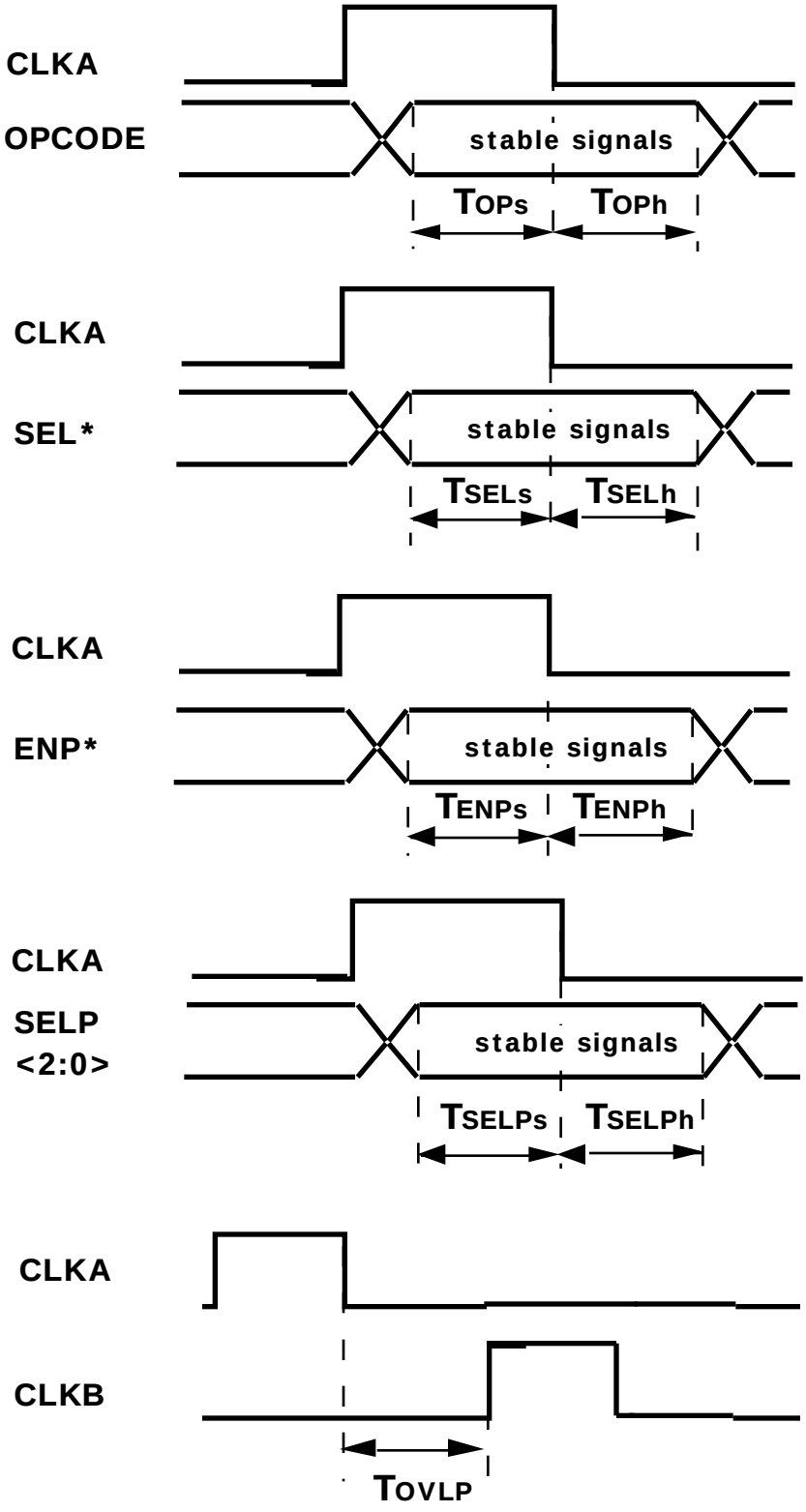
OpCode	Mean	Min	Max
Init	1.080	1.059	1.192
Reset Handshake	1.083	1.059	1.205
Clear Counter	1.021	0.995	1.158
Write	1.244	1.237	1.252
Read	12.611	12.044	12.912
Clear Hits	1.323	1.314	1.333
Input plane #0	15.304	15.183	15.395
Input plane #1	15.307	15.179	15.405
Input plane #2	15.308	15.177	15.403
Input plane #3	15.311	15.178	15.410
Input plane #4	15.313	15.192	15.410
Input plane #5	15.382	15.249	15.483
Count	1.747	1.738	1.756
Output	1.468	1.461	1.475
Nop0	1.074	1.064	1.090
Nop6	1.005	0.996	1.032
Nop13	0.974	0.963	1.019
Nop11	0.976	0.965	1.041
Nop14	0.977	0.960	1.056
Nop15	0.948	0.932	1.042
Shift	4.178	4.130	4.258

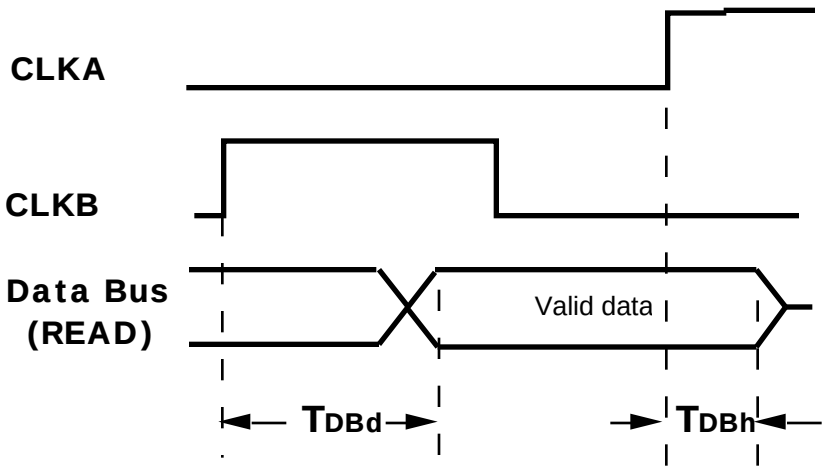
10 MHz

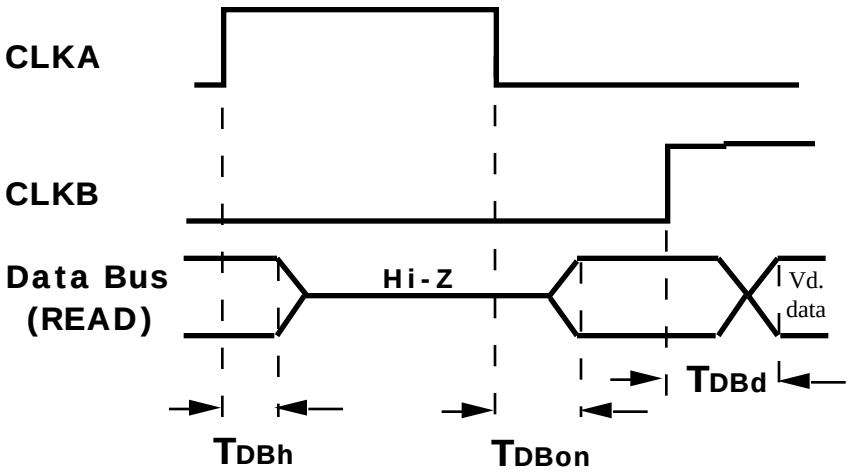
OpCode	Mean	Min	Max
Init	1.780	1.750	1.890
Reset Handshake	1.779	1.740	1.900
Clear Counter	1.655	1.630	1.780
Write	2.082	2.070	2.100
Read	25.213	23.900	25.850
Clear Hits	2.283	2.260	2.300
Input plane #0	32.746	32.410	32.960
Input plane #1	32.753	32.410	35.980
Input plane #2	32.755	32.400	32.980
Input plane #3	32.762	32.420	32.990
Input plane #4	32.770	32.430	32.990
Input plane #5	32.917	32.570	33.140
Count	3.097	3.080	3.120
Output	2.503	2.490	2.520
Nop0	1.710	1.690	1.720
Nop6	1.640	1.630	1.660
Nop13	1.609	1.600	1.650
Nop11	1.611	1.600	1.680
Nop14	1.612	1.600	1.690
Nop15	1.579	1.550	1.670
Shift	8.125	8.050	8.180

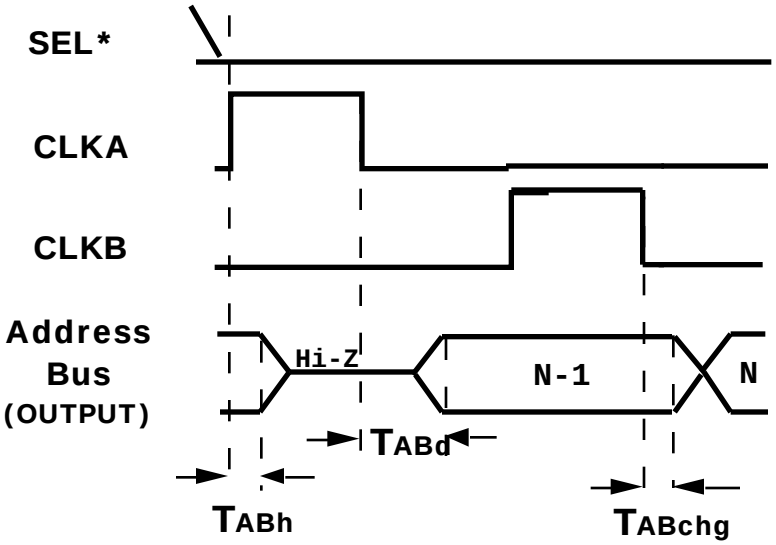
15 MHz

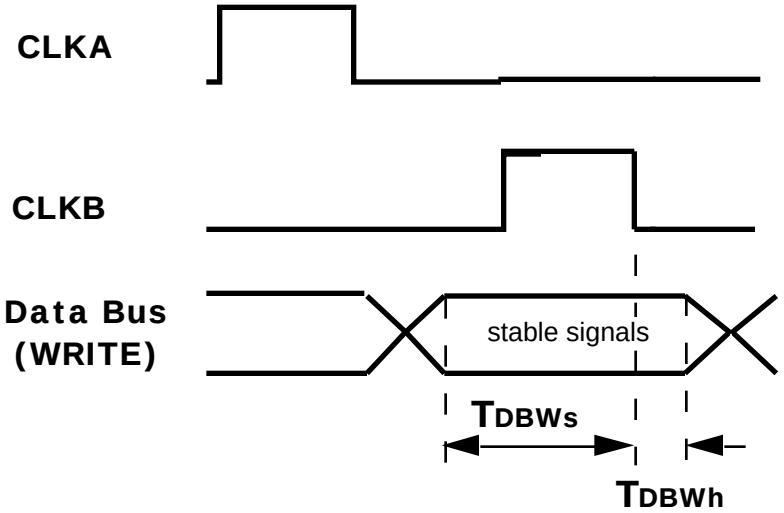
OpCode	Mean	Min	Max
Init	2.454	2.430	2.560
Reset Handshake	2.454	2.420	2.570
Clear Counter	2.270	2.240	2.410
Write	2.906	2.890	2.920
Read	36.751	34.790	37.840
Clear Hits	3.215	3.190	3.240
Input plane #0	47.266	46.860	47.540
Input plane #1	47.270	46.860	47.570
Input plane #2	47.274	46.870	47.570
Input plane #3	47.279	46.860	47.580
Input plane #4	47.290	46.880	47.590
Input plane #5	47.500	47.080	47.800
Count	4.425	4.400	4.450
Output	3.507	3.490	3.530
Nop0	2.323	2.310	2.340
Nop6	2.257	2.250	2.280
Nop13	2.228	2.210	2.270
Nop11	2.228	2.210	2.290
Nop14	2.229	2.210	2.310
Nop15	2.201	2.170	2.290
Shift	11.931	11.820	12.000

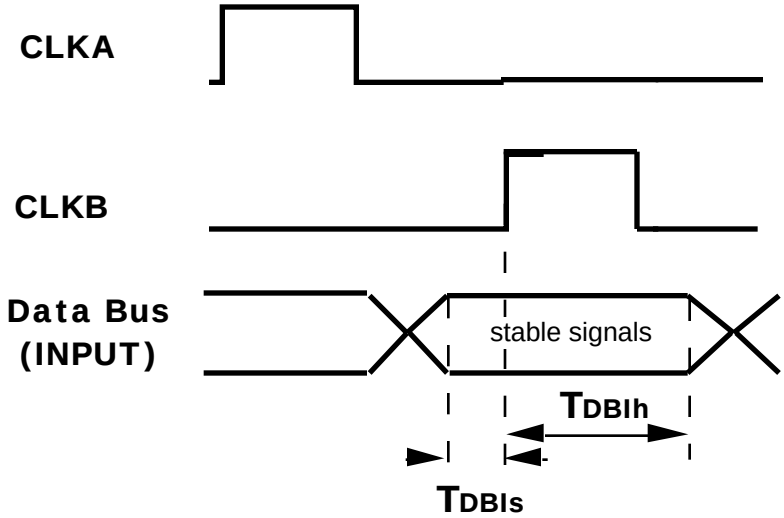

CLKA
OPCODE
stable signals
TOPs
TOPh
CLKA
SEL*
stable signals
TSELs
TSELh
CLKA
ENP*
stable signals
TENPs
TENPh
CLKA
SELP
<2:0>
stable signals
TSELPs
TSELPh
CLKA
CLKB
TOVLP

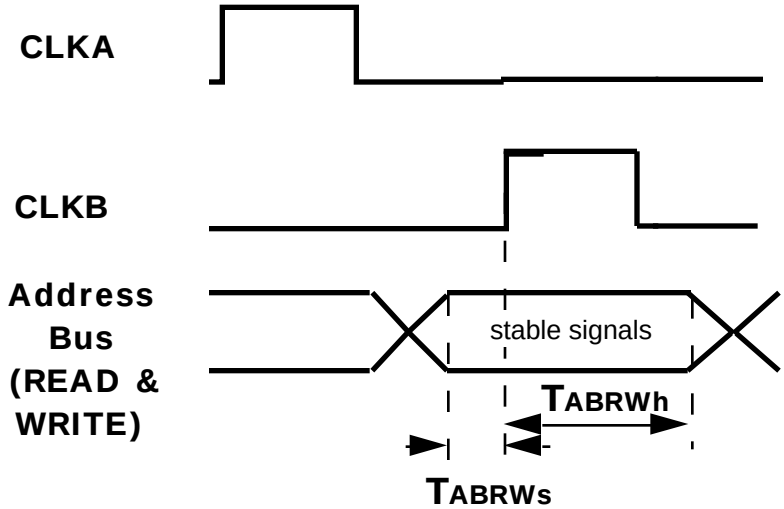

CLKA
CLKB
Data Bus
(READ)
Valid data
TDBd
TDBh

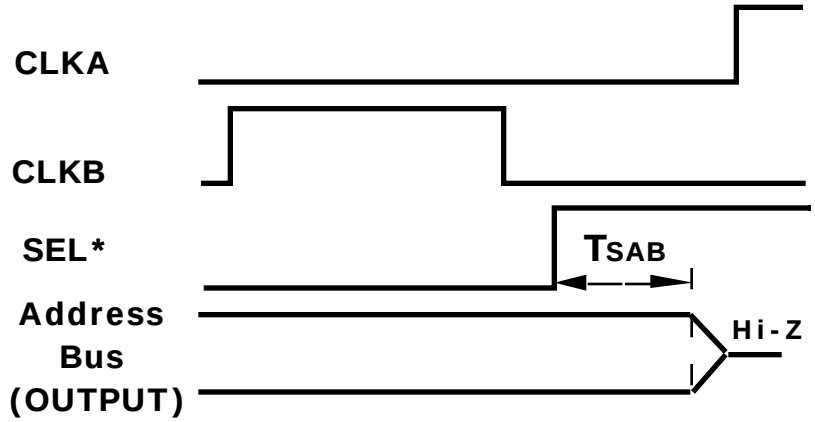

CLKA
CLKB
Data Bus
(READ)
Hi-Z
Vd.
data
TDBd
TDBh
TDBon

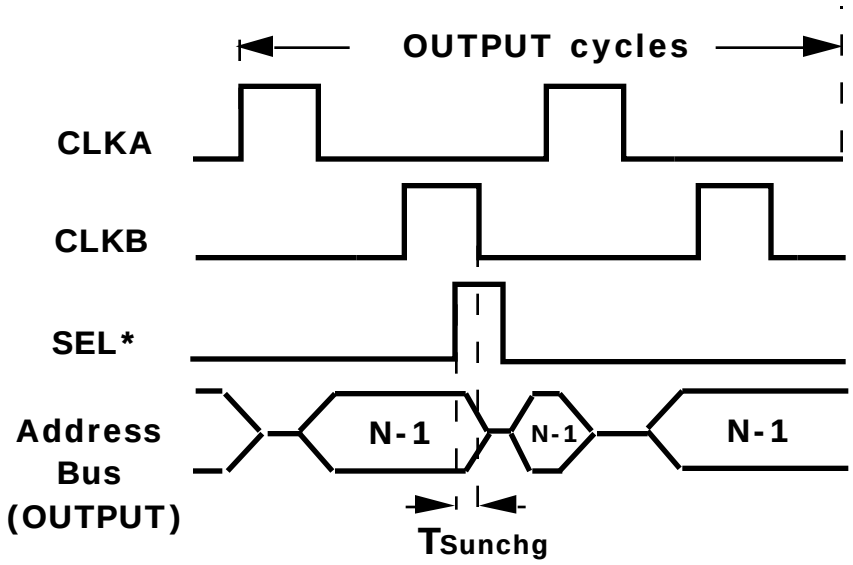

SEL*
CLKA
CLKB
Address
Bus
(OUTPUT)
Hi-Z
N-1
N
TABd
TABh
TABchg

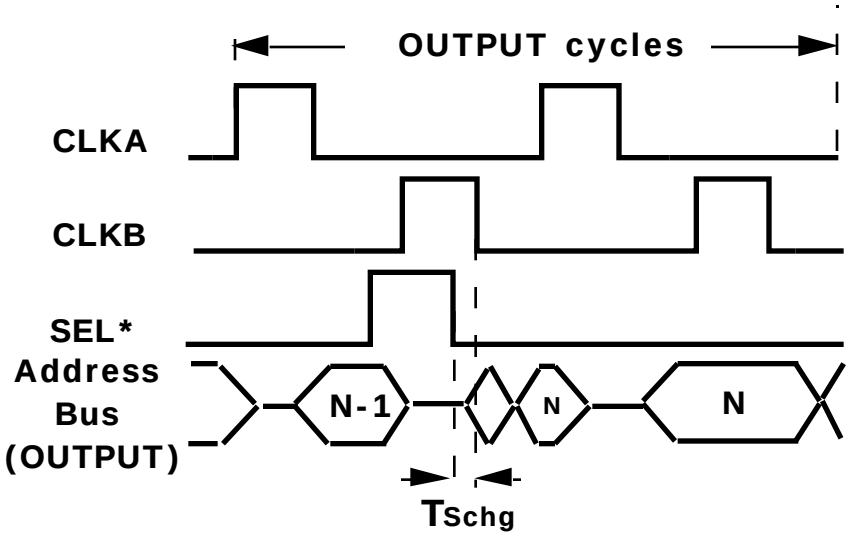

CLKA
CLKB
Data Bus
(WRITE)
stable signals
TDBWs
TDBWh

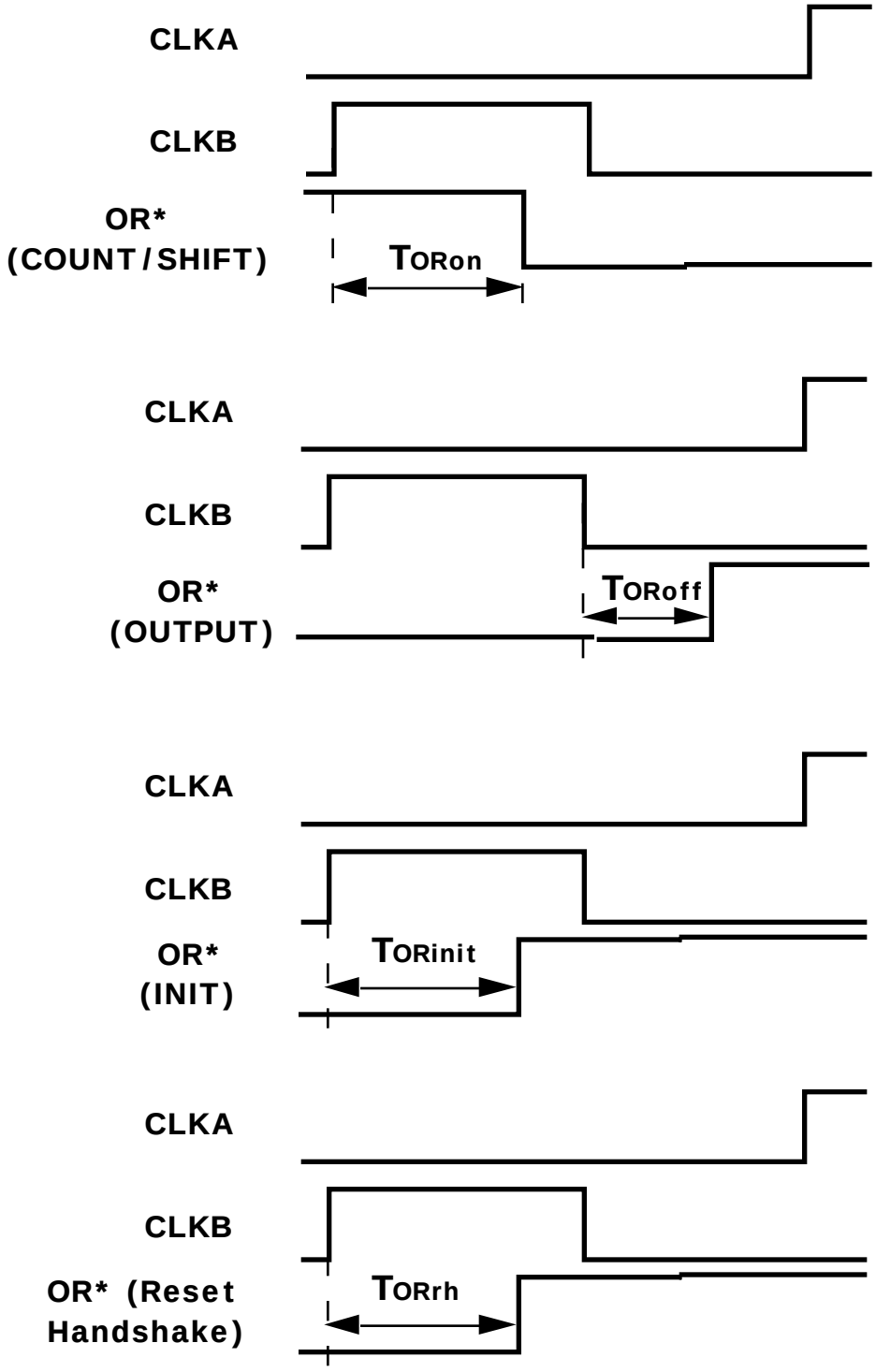

CLKA
CLKB
Data Bus
(INPUT)
stable signals
TDBIh
TDBIs

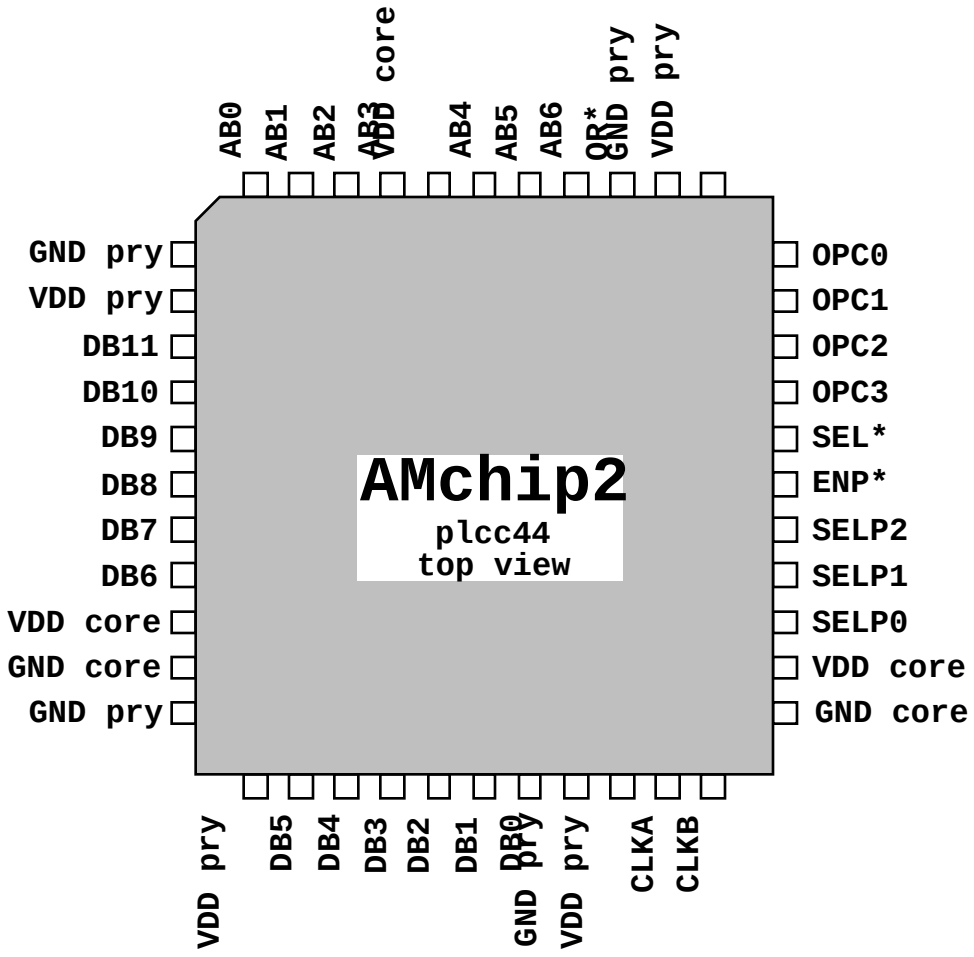

CLKA
CLKB
Address
Bus
(READ &
WRITE)
stable signals
TABRWh
TABRWs


CLKA
CLKB
SEL*
TSAB
Address
Bus
(OUTPUT)
Hi-Z


OUTPUT cycles
CLKA
CLKB
SEL*
Address
Bus
(OUTPUT)
N-1
N-1
N-1
TSunchg


OUTPUT cycles
CLKA
CLKB
SEL*
Address
Bus
(OUTPUT)
N-1
N
N
TSchg


CLKA
CLKB
OR*
(COUNT/SHIFT)
TORon
CLKA
CLKB
OR*
(OUTPUT)
TORoff
CLKA
CLKB
OR*
(INIT)
TORinit
CLKA
CLKB
OR* (Reset
Handshake)
TORrh


AMchip2
plcc44
top view
AB0
AB1
AB2
AB3
VDD core
AB4
AB5
AB6
OR*
GND pry
VDD pry
GND pry
VDD pry
DB11
DB10
DB9
DB8
DB7
DB6
VDD core
GND core
GND pry
OPC0
OPC1
OPC2
OPC3
SEL*
ENP*
SELP2
SELP1
SELP0
VDD core
GND core
VDD pry
DB5
DB4
DB3
DB2
DB1
DB0
GND pry
VDD pry
CLKA
CLKB

Appendix B.

GLUE CHIP Architecture

Each level of the Glue chip is implemented as a number of identical GLUE chips by using Field Programmable Gate Arrays (FPGA). The final designs of the GLUE chips were implemented in VHDL for the Warp compiler . The VHDL code and outputs of WARP compiler and place&route tools can be found in:

suncdf1.pi.infn.it:meschi/glulib_v1/vhdl.

Each level of GLUE is in a different subdirectoty. Current version for Glue0 is in *./bg110clk_july_97*, for GLUE1 and GLUE2 are in *./cglue1* and *./cglue2* respectively.

The Glue chip is comprised of four main sections *Input Data Cell*, *Input Layer Cell*, *Input Opcode cell* and *Output Engine* (see figure 1). The GLUE0 chip is comprised additional section *Clock Generator* for clock distribution and multiple phase generation for AM chips.

Input Data Cell

During normal SVT operation (MODE = PIPE) the *Input Data Cell* is a two-stage pipeline which synchronizes the DATA_UP lines from the upper level GLUE1 chip using two consecutive rising edges from CLK1 and CLK3 (fig.1). This is necessary because during the INPUT phase DATA_DW must be valid at the AM chip on the falling edge of CLKB. CLKB falling edge delay with respect to CLK3 rising edge must be adjusted to satisfy the setup times at the AMCHIP inputs. IN MODE=READ, DATA_DW is sent to DATA_UP directly. In MODE=WRITE, conversely, DATA_UP is sent to DATA_DW.

Input Layer Cell

The *Input Layer Cell* synchronizes LAYER_UP to the downstream logic. When the operating mode is PIPE (i.e. during normal SVT operation) the layer number is registered and fed to the AM chips. It must be valid before the falling edge of CLKA. During VME operations (Read or Write) input from the higher level GLUE is copied directly to the output.

Input Opcode cell

The *Input Opcode cell* synchronizes the upstream Operating Code OPC_UP to the falling edge of CLKA, when the operating mode is PIPE (i.e. during normal SVT operation). The 4-bit OPC_UP input bus is registered at the rising edge of CLK1. And is then ready for output onto the OPC_DW lines. During VME operations (Read or Write) the input is copied directly to the output.

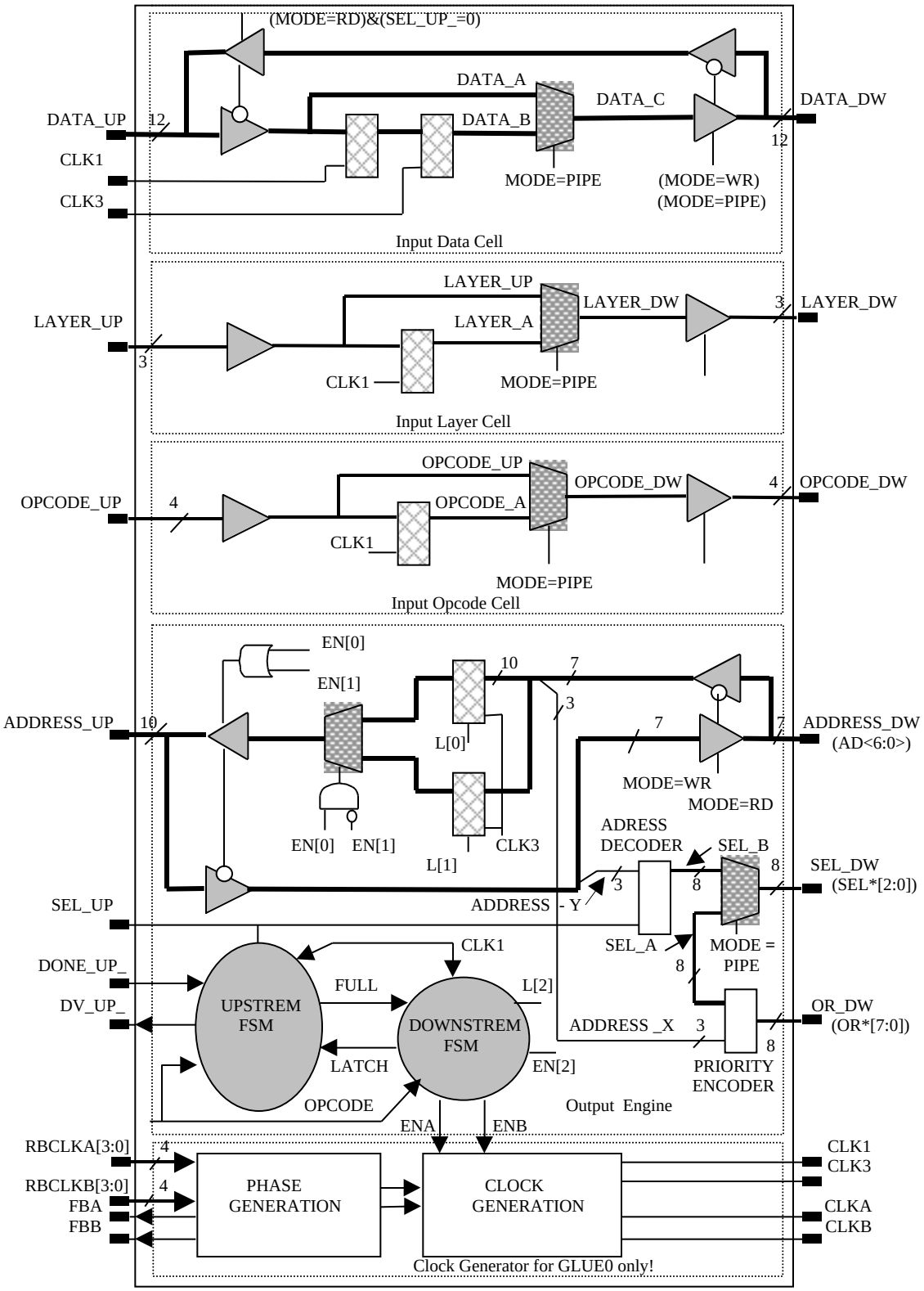
Output Engine

The *Output Engine* GLUE0 handles the data stream to and from the AM chips:

- ✓ Controls output pipeline,
- ✓ Coordinates input/output to or from the various down stream AM chips, by encoding/decoding upstream address
- ✓ Enabling Read or Write to a single AM chip or GLUE branch,
- ✓ Prioritizing outputs from the various leaves/branches connected to the GLUE0.

The main parts of the *Output Engine* are:

- ❑ **The Double Address Register**



Appendix B Figure 1. GLUE0 Chips internal Structure.

- **The Priority Encoder**
- Two finite state machines, one for down (The **Downstream FSM**) and one for the up side (The **Upstream FSM**) GLUE0 chip.

The Double Address Register during normal SVT operation (MODE = PIPE) takes its input from ADDRESS_DW and outputs to ADDRESS_UP(see fig.?). The two registers are enabled in turn by the L[0:1] signals from the Downstream FSM. In the registers the most significant bits of the address (ADDRESS_X) are merged in from the Priority Encoder. Each of the two registers is alternately output under the control of the EN[0:1] signals from the Upstream FSM. During VME operation the Double Address Register transparently copies ADDRESS_UP least significant bits to ADDRESS_DW.

The Priority Encoder generates enable signals (Sel_A[7:0]) for the AM chips based on the Output Ready (OR_DW) values. The SEL_A[7:0] signals are used to enable output from a given AM chip (when OPCODE = OUTPUT) by sending them out via SEL_DW_[7:0]. During operations other than OUTPUT the Address Decoder drives the SEL_DW [7:0] bus.

The Upstream FSM for Glue0 coordinates the Address Register pipelining through signals L[0:1] (latch the corresponding register), and EN[0:1] (enable output of the register). State bits F[0:1] indicate that the corresponding register (A and/or B) is full. When both registers are full the machine asserts the FULL signal to the Downstream FSM which inhibits further loading of data in the registers until the pipeline is emptied from above. In this case the PRI bit indicates which register was loaded first. The LATCH signal enables the loading of a register at next clock edge. Whereas the S = DONE_UP&SEL_UP signal frees a register at the next edge. The EN[0:1] signals enable the output of either register to the ADDRESS_UP pins.

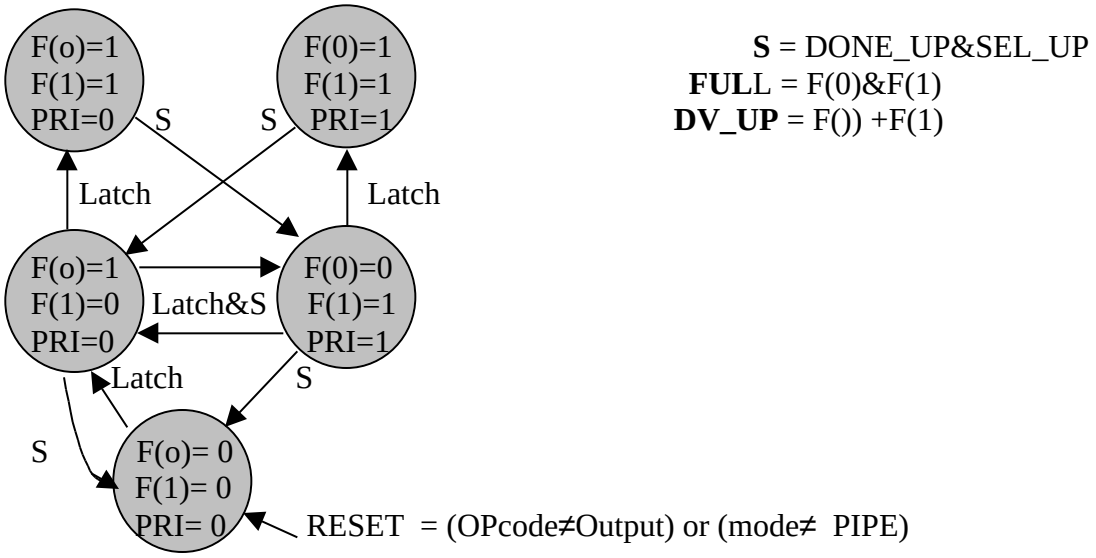
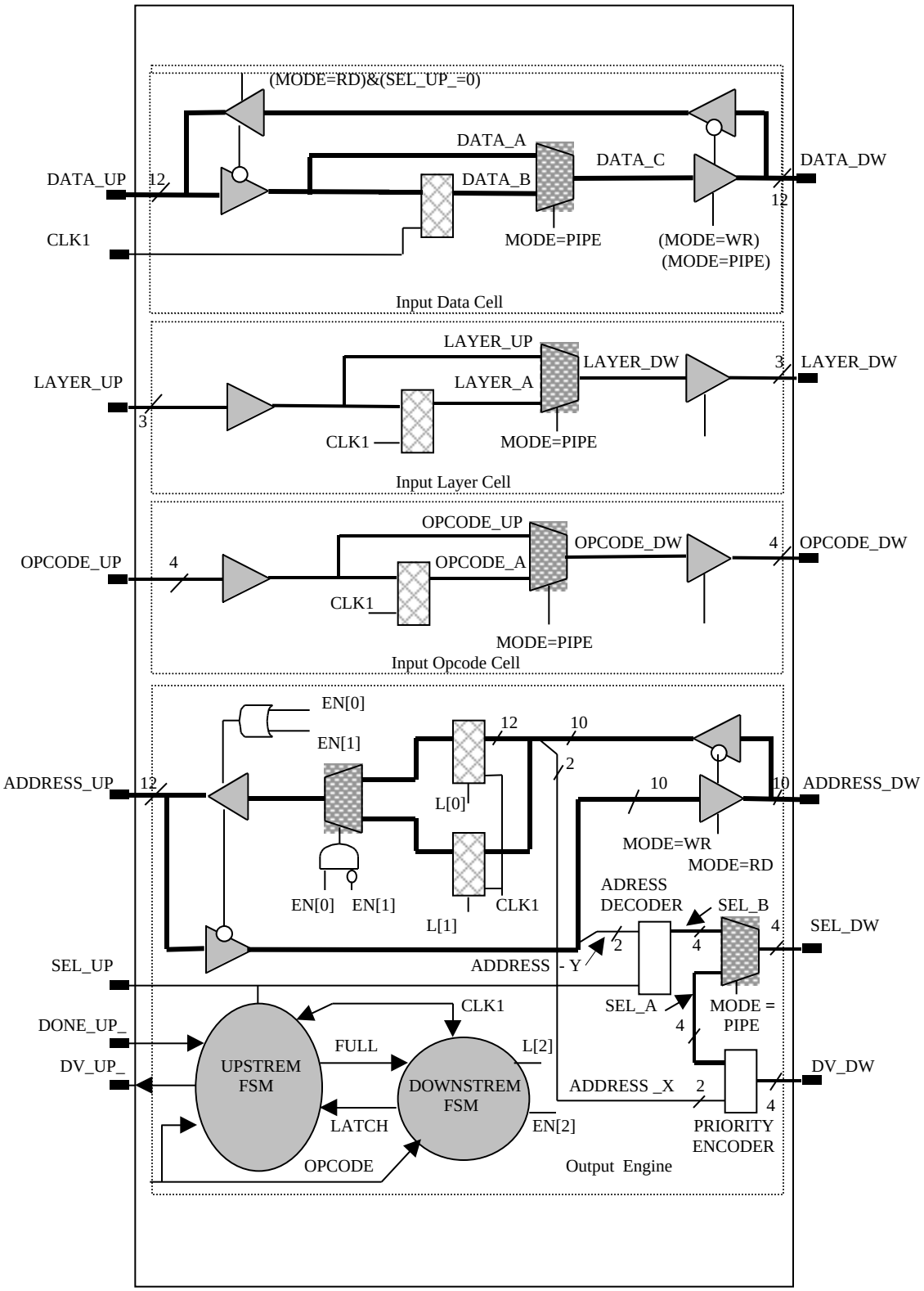
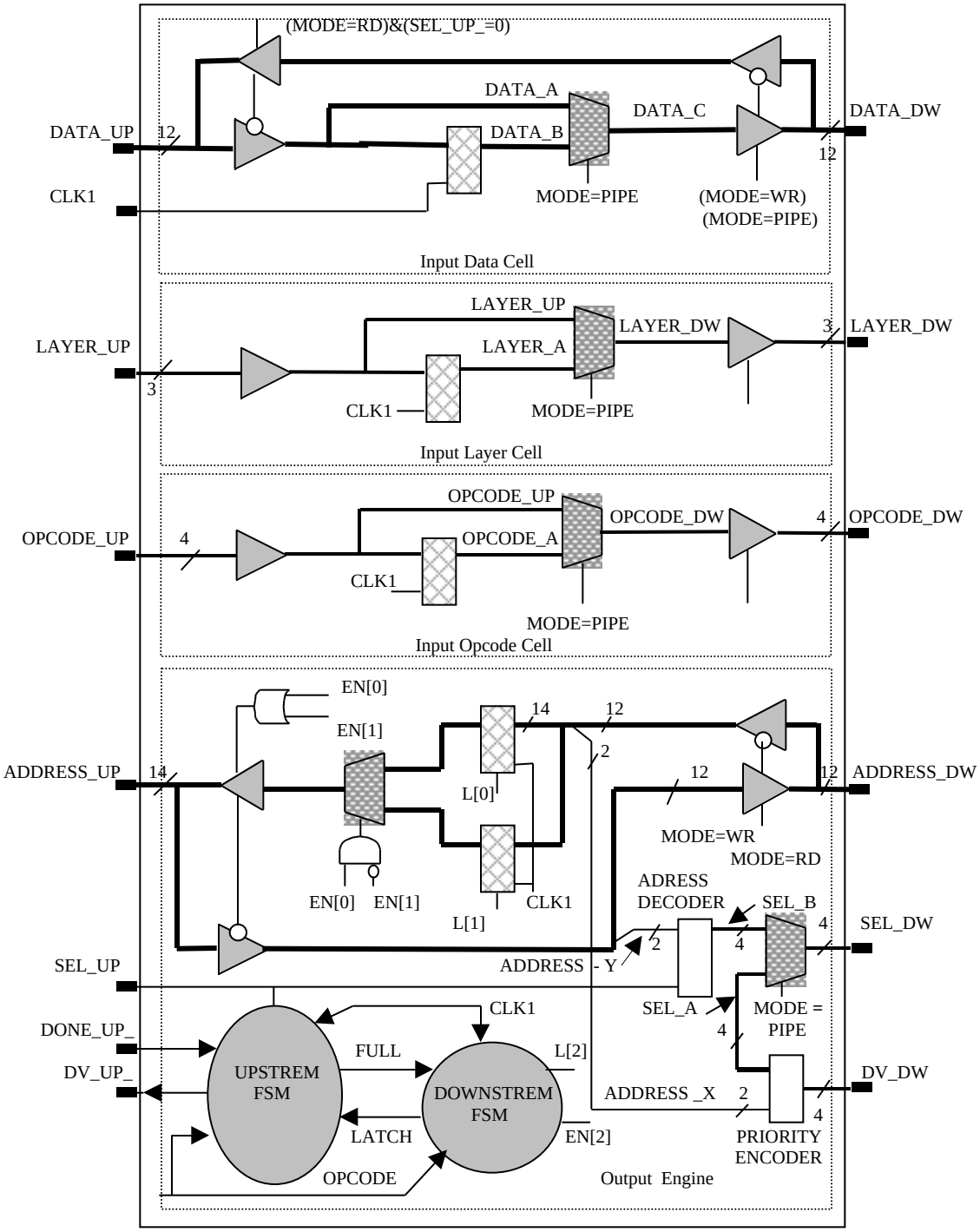


Figure 2. The GLUE0 Upstream Finite State Machine

The Downstream FSM coordinates the output from the various downstream AM chips or GLUE chips. The machine stays inactive when the *Halt* mode is invoked from the VME controller, otherwise it has two modes. In *Read* or *Write* mode there is no pipeline to control, the machine stays in S0 with both CLKA and CLKB enabled to the



Appendix B. Figure 2. GLUE1 Chips internal Structure.



Appendix B. Figure 3. GLUE2 Chips internal Structure.

AM chips. When mode = PIPE the machine is activated by an Output Opcode. This disables CLKB and waits for an output ready (OR) from the AM chips (S1). When a valid address is ready in the AM chips CLKA is also disabled and the machine listens for the pipeline full (FULL) signal (S2). FULL is deasserted from the UpStream FSM to indicate that a new address can be latched and so CLKB is enabled to tell the AM chips to present the address at the input (ADDRESS_DW) (S3). At the next cycle the LATCH

flag is asserted to instruct the Upstream FSM to latch the new data (S4) and then the machine goes back to listen for a new valid address in (S2). In a standard Output cycle the machine cycles through S2-S3-S4 in three clock cycles.

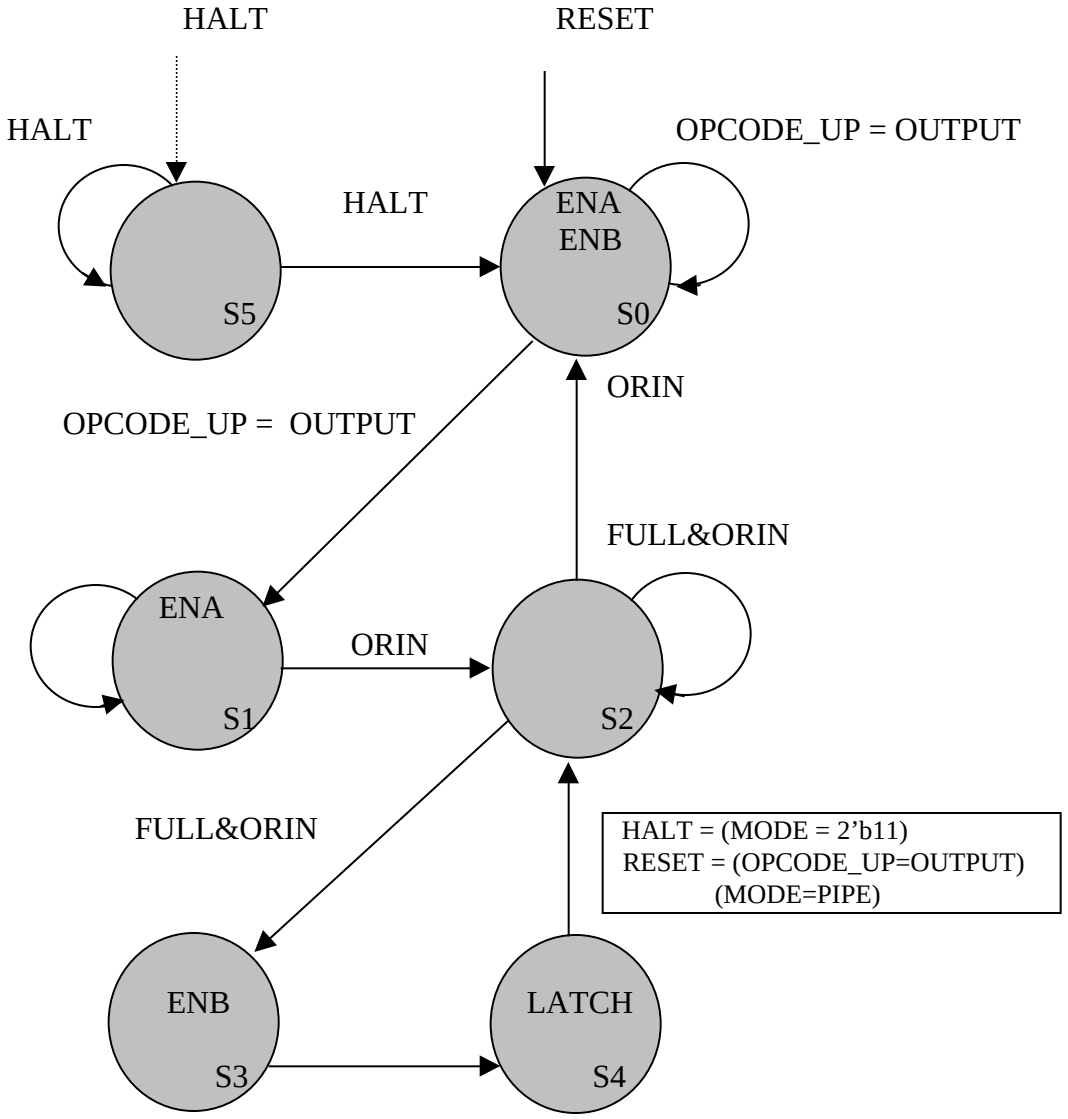


Figure 5: The GLUE0 Downstream Finite State Machine.

The GLUE2 chip has architecture identical to a GLUE1 except for the address bus widths, i.e. the ADDRESS_DW bus is 12 bits wide and the ADDRESS_UP bus is 14 bits wide (see figure 4). There are more differences between GLUE 1 and GLUE0 (see figure 2,3):

- ✓ 1. The ADDRESS_DW bus is 10 bits wide and the ADDRESS_UP bus is 12 bits wide;
- ✓ 2. There is no clock generation and the entire chip works on one clock signal (CLK1), therefore the Input Data Cell and Address Register1 both use CLK1.

- ✓ 3. The four Data Valid (DV_), inputs from the four GLUE)s are prioritized to 4 SEL_A[0:3} signals that can be sent (OPCODE=OUTPUT) to Sel_DW_ in alternative to the SEL_B[0:3] signals from the decode of the ADDRESS_UP 2 most significant bits.
- ✓ 4. The Downstream Finite State Machine is simplified to only 3 states.

Appendix C

VME operations:

- *OPC* operations: all Amchip opcodes operations
- *Special* operations:
  1. Read register 0
  2. Read prom
  3. Switch from P3 mode to VME mode
  4. Switch from VME mode to P3 mode
  5. Switch from Halt mode to Pipe mode

VME Addresses map:

VME Address	Function
ADD<31…27>	Geographical address
ADD<26..24>	Assumed as part of geoadd, **board is selected only if equal to 0**
ADD<32..2>	Internaly mapped addresses
ADD<1..0>	Not used in A32D32 transfers

➢ *The slave vme interface operation to be performed is mapped in ADD<23..19>*

OPC operations have ADD<19> set to 1, *Special* operations have ADD<19> set to 0.
OPC operations have opcode bits OPC_UP<3..0> mapped in ADD<23..20>.
Special operations which require a write cycle always set to 0 OPC_UP<3..0>.

Table 1. Valid operations and related addresses.

VME ADDRESS <23..0>		Function
begin	end	
0x000000	0x07FFFC	Read Reg0
0x080000	0x0FFFFC	NOP0
0x100000	0x17FFFC	Read Prom
0x180000	0x1FFFFC	Read
0x200000	0x27FFFC	P3 VME
0x280000	0x2FFFFC	Output
0x300000	0x37FFFC	VME P3
0x380000	0x3FFFFC	Write
0x400000	0x47FFFC	Halt Pipe
0x480000	0x4FFFFC	Input
0x580000	0x5FFFFC	Clear Hit Register
0x680000	0x6FFFFC	Nop6
0x780000	0x7FFFFC	Shift
0x880000	0x8FFFFC	Count
0x980000	0x9FFFFC	Clear Counter
0xA80000	0xAFFFFC	Init
0xB80000	0xBFFFFC	Nop11
0xC80000	0xCFFFFC	Reset Handshake
0xD80000	0xDFFFFC	Nop13
0xE80000	0xEFFFFC	Nop14
0xF80000	0xFFFFFC	Nop15

Table 2. How internal addresses ADD<18...2> are used in the board.

Function	Addresses <18..2>																
	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2
Read Reg0				reserved													
Nop0																	
Read Prom	X											**ADD_PROM<15..0>**					**mandatory!**
Read	**Layer_UP**											**ADD_UP<13..0>**					**mandatory!**
P3 VME																	
Output																	
VME P3																	
Write	**Layer_UP**											**ADD_UP<13..0>**					**mandatory!**
Halt Pipe				reserved													
Input																	
Clear Hit Register																	
Nop6																	
Shift																	
Count																	
Clear Counter																	
Init																	
Nop11																	
Reset Handshake																	
Nop13																	
Nop14																	
Nop15																	

Table3. VME Data for the various operations.

Function	**Data <31..0>**																															
	D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21			D18	D17	D16	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Read Reg0										BRD_SEL	TMODE		MODE1	MODE0																		
Nop0																																
Read Prom	EPROM DATA																			**X**												
Read																																**AM DATA**
P3 VME																	X															
Output	**Out address**														OUTDV																	
VME P3																	X															
Write											**X**																					**AMDATA**
Halt Pipe																	X															
Input											X																					**AM DATA**
Clear Hit Register																	X															
Nop6																	X															
Shift																	X															
Count																	X															
Clear Counter																	X															
Init																	X															
Nop11																	X															
Reset Handshake																	X															
Nop13																	X															
Nop14																	X															
Nop15																	X															