

The AM++ Board for the Silicon Vertex Tracker upgrade at CDF

A. Annovi^{2,3}, A. Bardi³, M.Bitossi^{2,3}, R. Carosi³, M.Dell'Orso^{2,3}, P. Giannetti³, P.Giovacchini^{2,3},
M.Piendibene³, F. Schifano¹, B. Simoni², F.Spinella³, S. Torre^{3,4}, R. Tripiccione¹,

¹Physics Dept and INFN Ferrara, Italy

²Dipartimento di Fisica, Università di Pisa, Pisa, Italy

³INFN Pisa, Italy

⁴Dipartimento di Fisica, Università di Siena, Siena, Italy

Abstract

The main effort of the Silicon Vertex Tracker upgrade is the production of a new Associative Memory (AM), ~40 times larger than the existing one. The Associative Memory is the SVT part dedicated to the pattern recognition specific task.

We have designed a new Associative Memory Board (AM++) to handle 128 standard cell Associative Memory chips. We describe the design, the tests and the performances of the new AM++ for CDF.

I. INTRODUCTION

Dedicated hardware systems, based on the use of an Associative Memory (AM) [1,2], have been proposed for track recognition in high-energy physics experiments.

A programmable AM based on a modular architecture has already been realized using full custom ICs [2] and Field Programmable Gate Arrays [3].

We have designed a new Associative Memory Board (AM++) for standard cell associative memory chips [4] dedicated to the Silicon Vertex Trigger (SVT) [5] upgrade at CDF. The new AM system increases the SVT computational power for high luminosity running. The SVT upgrade was built in a very short time, since it is designed to require a minimum of new hardware. The new AM++ boards themselves have been under design for a while based on R&D work that was done for the LHC, the Fast Track processor (FTK) [6]. The new CDF board compared to the FTK version, is enriched of important features for the standard cell chip operation control and for diagnostics. However the AM++ has a reduced input bandwidth. In FTK, in fact, six input buses are provided for hits coming from the detector. For a 6 layer pattern recognition each layer is provided of a private bus. In the old SVT structure, instead, hits from all the layers are multiplexed on a single bus and the CDF AM++ board complies with this standard.

III. HARDWARE DESCRIPTION

The AM++ is a 9U VME board working at a clock frequency of 40 MHz. It has a modular structure, consisting of 4 smaller boards, the Local Associative Memory Banks (LAMBs). Each LAMB contains up to 32 Associative Memory chips (Amchips), 16 on each side of the board. The SVT upgrade uses only the 16 AMchips on the top face of the LAMB, to reach the required pattern density (512k patterns

per wedge). The AM++ board is sketched in Figure 1. The structure of the LAMB is also shown. The AM chips come in PQ208 packages and contain the stored patterns, memory, pattern matching and read-out logic.

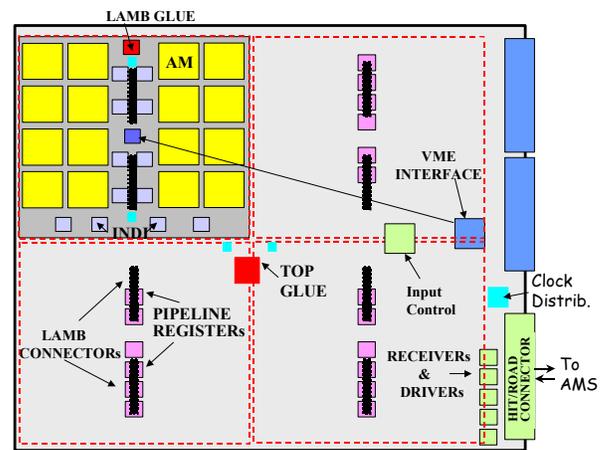


Figure 1: The AM++ Board

The AM must be able to store all trajectories of interest and extract the ones compatible with a given event. A trajectory is compatible with an event if all (or a majority of) detector channels crossed by that trajectory have fired in that event. In order to contain the trajectory memory within an acceptable size the AM operates at a coarser resolution than the actual detector. This is usually done by clustering single contiguous detector channels into larger superstrips. In the following, we will call “hits” the addresses of fired superstrips in each layer, and “roads” the coarse resolution trajectories (each road corresponds to an array of superstrip addresses – one per detector layer).

The AM finds roads linking a drift chamber track (XFT layer) and the requested number of silicon hits. Five silicon layers are used.

When an AM++ board starts to process an event, the hits are received by the Input Control chip. Hits of different layers, coming from the P3, are mixed on a single bus, as required by the SVT architecture. However inside the Input Control chip they are multiplexed into 6 buses, one per layer, and simultaneously sent to the four LAMBs. So the LAMBs maintain the larger input bandwidth realized in the FTK project and can be used also in more complex future applications.

As soon as hits are downloaded into the LAMBs, the locally found roads set the request to be read out (Data Ready). Once the event hits are completely read out, the LAMBs make the last matched roads available within a few clock cycles.

The outputs of the 4 LAMBs are multiplexed into a single bus by a purposely-designed TOP GLUE chip sending roads to the AM Sequencer (AMS) through the P3 connector.

The TOP GLUE, strobed by the AMS, distributes to the LAMB Glues a set of Operation Codes (OPCODEs) that define the road readout order. The majority logic available in the AM chip is driven so that found roads are ordered giving priority to the most constrained matches.

III. LAMB

Six hit buses, one for each detector layer, are fed in parallel to the four LAMBs and distributed to the 32 AM chips on the LAMB through 12 fan-out chips called Input Distributors (INDI). For the output road address bus, the AM chips on each LAMB are divided into 4 pipelines, each corresponding to a row in Figure 2, consisting of 8 chips, 4 on the front and 4 on the back. Each AM chip receives an input bus where road addresses found in previous chips are propagated, multiplexes it with the road addresses internally found in that chip, and sends the output to the next chip. Signals propagate from one chip to another on the top and bottom PCB planes through very short pin-to-pin lines, without any vias.

The outputs of the 4 pipelines are then multiplexed into a single bus by a specially designed GLUE chip on each LAMB.

The LAMB board has been designed, optimized, simulated, placed, and routed with Cadence software. It represents a significant technological challenge due to the high density of chips allocated on both sides of the board and the use of the advanced Chip-scale packages (CSP) for the 12 INDI chips and the GLUE chip.

Successful operation has been tested at a clock frequency of 40 MHz using FPGA and standard cell associative memories pin compatible devices.

IV. AM++ CONTROL

The internal AMchip [4] configuration specifies a default matching criterion that is activated at the beginning of each event. The matching criterion is controlled through 2 main parameters, constraining the pattern matching algorithm: *THR* and *required_layers* [4]. *THR* is the number of layers that are required to be matched in a road in order to get a road match. The *required_layers* option is a 2-bit word. Each bit activate the option for one out of two predefined layers. The road match can be generated only if all active *required_layers* have a match. We use the *required_layer* flag to force the matching of the XFT layer.

The matching criterion can be changed during the event processing through the OPCODEs [4]. The following is the list of operations the GLUE executes for each event, in response to the OPCODEs received from the controller.

1. *Init*: the AM++ and all LAMBs are reset just after the end of previous event (OPCODE=5 from the AM++ controller

strokes the Init function). The default matching criterion is loaded (THR and *required_layers* default values).

2. *Input* (OPCODE=4, event hits loading): hits are received from the P3 connector and fed to INDI chips to be propagated to the AMchips.
3. *Decrease Threshold* (OPCODE=7). It is executed when the Top GLUE receives the End-of-Hit (OPCODE=1) from the AM++ controller. When the roads corresponding to the current matching criteria are all readout, the Decrease Threshold opcode is distributed to the AMchips in order to change the value of the THR parameter.
4. *Output* (road readout): roads can be readout during or just after the *Input* phase. The matching criterion must not be changed (i.e. OPCODEs must not be sent to AMchips), until the last incoming hit has arrived.

The *Input* operation does not use the busses necessary for roads readout (*Output* operation). Hence, the *Input* and *Output* functions can overlap. With this option the last road would leave the LAMB few clock cycles after the last hit gets in. In fact roads fired in all layers (5/5: 5 matches for 5 silicon layers) can be collected from the AMchip pipelines as soon as they are available. They do not need to wait until the *input* phase is finished. Roads are thus partially processed even before the event readout is complete. Roads characterized by a small inefficiency (4/5: 4 matches for 5 silicon layers), instead, are collected outside the system only when the *input* phase is finished. Roads can be ordered (5/5 first, 4/5 second) or can be readout without order, at the end of the *input* phase, as it was in the past. The ordered readout is obtained setting the default THR parameter to 6 (5 silicon layer matches + XFT match) and decreasing the THR parameter to 5 when the 5/5 roads have been exhausted and the *input* phase is finished. If the old system has to be reproduced, the order is destroyed setting the default THR to 7 out of 6 layers, and decreasing it directly to 5 when the End-of-Hit is received from the controller.

V. THE GLUE LOGIC

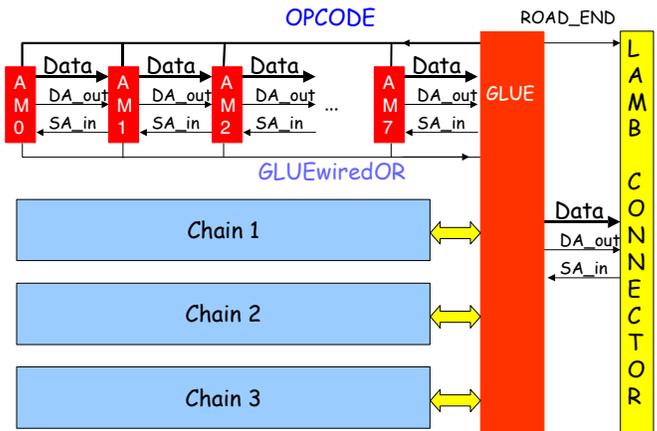


Figure 2: Device organization on the LAMB. The GLUE chip collects roads from four AMchip pipelines. Each chain can allocate a maximum of 8 chips. Roads from the four chains are merged in a single output sent to the Top GLUE on the AM++

Figure 2 shows how the different devices are organized on the LAMB. The GLUE chip collects roads from four AMchip

chains and controls the matching criterion (all the control signals shown in the figure are active low).

The GLUE chip logic is organized to move roads along the pipelines as soon as possible to reduce latency. The AMchip pipelines can be so long (up to 8 chips) that a road matched in a chip far from the GLUE would take many clock cycles to be collected outside the LAMB. For this reason the GLUE has four independent engines to control the four pipelines in parallel. If a pipeline is ready to change matching criterion, it can be changed in advance, even if other pipelines are still working on the default criterion. This makes the GLUE chip much more complex (roads belonging to different criteria cannot be mixed in the GLUE output), but helps to build a continuous stream of roads up to the P3 connector.

On each Data bus, data are organized into packets of 2 words: the road address (Road-ADD), and the bitmap, the list of the matched superstrips inside the road. This is a 6 bits word, one per layer, that contains information about the fired layers: a bit equal 1 (0) means that the corresponding layer was (was not) fired.

Packets are pushed ahead in the AM chip chains (see figure 2), up to the GLUE. The transfer is controlled by two handshake signals between two adjacent chips: the *Data Available* (DA_out) from the upstream chip to the receiving one and the *Space Available* signal (SA_in) sent back by the receiving chip to the upstream chip.

The protocol is the following: a 2 word packet transfer is started when both the DA_out and the SA_in signals are asserted. The DA_out signals the availability of a couple of words. The SA_in signals the availability of enough space to receive the 2 words. The first word (Road-ADD) is latched in the input register on the clk rising edge on which both DA_out and SA_in are asserted. The second word (bitmap) is latched on the next clk cycle regardless of the state of control signals.

A simplified scheme of the GLUE chip is shown in figure 3. The first purpose of the GLUE is to multiplex into a single stream the 4 AM chain busses.

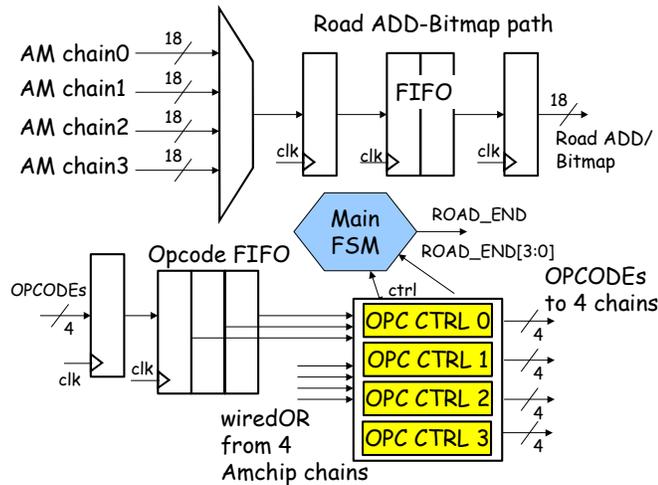


Figure 3: The Glue logic.

The upper part of figure 3 shows that after the multiplexer there is a register, then a 2 word FIFO and finally the output

register. This means that each word needs at least 3 cycles to go from input pins to output. The controlling FSM makes sure that packets composed of 2 words (Road-ADD, bitmap) are not broken.

The second purpose of the GLUE is to send the OPCODE lines to all the AMchips on the LAMB (see figure 2). Four buses are provided by the Glue, one for each AMchip chain. All the AMchips in a chain receives the OPCODEs in parallel. OPCODE and wired_DA (wired Data Available) lines allows the Glue to control the AMchip matching criteria. Each AMchip has an output called wired_DA. It is active if the AM chip has at least one road available for readout. The eight wired_DA signals in a chain are wired together into the GLUEwired_DA signal. The GLUEwired_DA is the logic OR of all wired_DA pins in a chain.

The GLUEwired_DA provides fast information to the GLUE on the availability of roads inside each of the four chains (eight AMchips per chain). The GLUE logic must take care of the fact that the wired_DA signal becomes active with some clk cycles of delay with respect to the incoming hits or OPCODEs that actually generates the road match. The GLUE guarantees that roads extracted in correspondence with different OPCODEs are not mixed.

The GLUE logic informs the TOP GLUE using the Road_end signal as soon as all found roads for a given criterion have been readout.

The bottom part of figure 2 shows the OPCODE logic. The OPCODEs coming from the TOP GLUE go through a register and then reach the Opcode FIFO. This FIFO stores all OPCODE words (up to three) that have not been sent to all chains yet. There are 4 "OPC CTRL" blocks working in parallel, one for each AM chain. All 3 registers of the Opcode FIFO are accessed in parallel by the "OPC CTRL" blocks. Each block sends OPCODEs to one chain of AMchip. The Main FSM coordinates the 4 blocks.

If an OPCODE is made of 2 words (OPCODE word + DATA word), the "OPC CTRL" block identifies those packets and avoids the separation of words.

Every OPCODE is sent (written into the output register) to the chain only when the previous OPCODE has been completely processed. An OPCODE is considered processed when all roads generated by the OPCODE itself have been extracted from the AMchip chain. This condition is verified checking the AND of these two conditions:

- At least N clk cycles elapsed since the OPCODE has been sent, where N is the number of cycles necessary for the OPCODE to activate the GLUEwiredDA signal (in case of found roads).
- No more roads are left in the chain. This is verified checking that the GLUEwiredDA signal is inactive. This signal, described before, tells us if there are roads left inside any chip of the chain.

When the OPCODE is completely processed a ROAD_END signal is sent from the "OPC CTRL" to the "Main FSM" and the next OPCODE, if any, is sent to the chain. The purpose of the ROAD_END signal is to tell the "Main FSM" that roads for the current OPCODE are finished.

With regards to OPCODEs the "Main FSM" has 3 jobs to perform:

- Make sure that roads extracted with different OPCODEs are not mixed together.
- Propagate a ROAD_END signal, when all the four chains have exhausted their roads, to the TOP GLUE on the AM board.
- Remove from the FIFO the OPCODEs already processed by all four chains.

In order to accomplish the 1st task the "Main FSM" holds (it does not produce the Space Available signal) all data streams but those coming from chains that are processing the earlier OPCODE.

For the 2nd task it will send the ROAD_END signal only after the last road corresponding to an OPCODE has left the GLUE.

VI. BOARD CONFIGURATION & DIAGNOSTICS

All the logic functions of the AM++ board are implemented on two kind of chips, Xilinx CPLDs [8] and Standard Cell Accociative Memory chips (AMchips).

All chips on the AM++ and the LAMB boards are fully configurable and testable using the Jtag interface implemented in the devices, following IEEE Std 1149.1-1990.

Jtag circuitry includes a standard interface through which instructions and test data are communicated between chips and a host, making it possible to download a bitstream to the chips to configure their internal logic, as well as to setup its I/O pins to specific logic levels through Boundary Scan Register. Conversely it is possible to upload a bitstream from the chip to gather information about logic level sampled at its I/O pins.

CPLDs used in the AM++ board are commercially available chips whose internal architecture is fully configurable accessing the built-in Jtag Interface. The configuration is retained in a flash memory inside the chip, which preserves them from loosing configuration upon power down. Due to the serial nature of Jtag data flow, several chips can be connected in daisy chain, and accessed for individual configuration and test. With a download cable connected to a PC and programming software the serial bitstream can be downloaded to the chips accessing dedicated connectors on the AM++.

The AMchips are Standard Cell custom devices, which implement a Jtag interface as well.

This interface gives access to internal configuration of the AMchip which includes Pattern memory Writing/Reading as well as Boundary Scan logic.

AMchips are interconnected on each Lamb in 8 separate daisy-chains of 4 devices each, to build a total of 32 individual daisy-chains throughout the whole AM++ (see Figure 4)

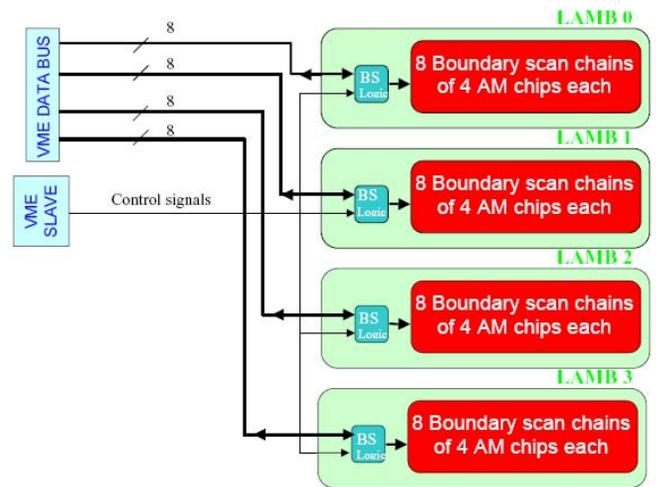


Figure 4: Daisy chains of AM chips on the LAMB. The 32-bit wide VME data bus gives access to the 32 chains in parallel. The custom VME Slave generates all the control signals necessary to correctly operate on the BSCAN interface of the AM chips.

The 32 separate chains are accessed in parallel by a VME slave which give access to all JTAG control signals of the chains, in parallel. The VME 32-bit wide data transfer allows us to program the 32 chains in parallel, so that time needed to store Patterns in the AMchips is a few seconds. Patterns are also checked using the boundary scan.

Very useful Boundary Scan diagnostics has been developed. These diagnostic tools are based on the use of two Boundary Scan instructions, SAMPLE/PRELOAD and EXTEST.

The SAMPLE/PRELOAD instruction allows us take a sample of the functional data entering and leaving the device while the device remains in its functional mode. This is useful to sample logic state of I/O pins. This instruction is also used to preload test data into a dedicated internal boundary-scan register, prior to loading an EXTEST instruction. During the EXTEST instruction, the boundary scan cells associated with outputs are preloaded with test patterns to test downstream devices. The input boundary cells are set up to capture the input data for later analysis.

The main diagnostic tool takes advantage of the Boundary Scan Instructions specified above and a special "Test Mode" implemented in the Input Control chip.

In this operating mode it is possible to write Hits data from VME Slave so that the Input Control chip sends these "test hits" to the 4 Lambs. Hit data flow to the INDI chips and are then sent to all the AM chips on a LAMB, in parallel.

In this way we could find interconnection problems in the hit flow. This test has been performed on all the produced AM++ and Lamb Boards to find assembling problems, before installation, and it is now part of the diagnostic tests on the experiment.

VII. CONCLUSIONS

A new associative memory board has been produced for the SVT upgrade in the CDF experiment. This board provides an improvement of a factor ~40 larger trajectory bank, which for SVT application translates to a road size 3 times smaller. A

higher working frequency is also possible. The AM++ has been successfully installed and used in CDF, running at 40 Mhz.

VIII. REFERENCES

- [1] M. Dell'Orso and L. Ristori, "VLSI structures for track finding", *Nucl. Instr. and Meth.*, vol. A518, 2004 pp. 532-536.
- [2] R. Amendolia et al., "The AMchip: a Full-custom MOS VLSI Associative memory for Pattern Recognition", *IEEE Trans. on Nucl. Sci.*, Vol. 39, 1992 pp. 795-797.
- [3] A. Bardi et al. "A Programmable Associative Memory for Track Finding", *Nucl. Instr. and Meth.*, vol. A413/2-3, 1998 pp.367-373.
- [4] A. Annovi et al. "A VLSI Processor for Fast Track Finding Based on Content Addressable Memories", submitted to this conference
- [5] W.Ashmanskas et al., "The CDF Silicon Vertex Tracker", *Nucl. Instr. and Meth.*, vol. A518, 2004 pp. 532-536.
- [6] A. Annovi et al. "A Pipeline of Associative Memory Boards for Track Finding", *IEEE Transactions on Nuclear Science*, Vol 48, No 3, 595,(2001).
- [7] IEEE Std 1149.1-1990 IEEE Standard Test Access Port and Boundary-Scan Architecture –Description. http://standards.ieee.org/reading/ieee/std_public/description/testtech/1149.1-1990_desc.html
- [8] Xilinx, The Programmable Logic Company (2001, Feb.). Xilinx Data Book 2000. Xilinx. San Jose, CA. [Online]. Available:http://www.xilinx.com/products/silicon_solutions/index.htm.

