Content-Addressable Memory (CAM) in Actel Devices

Introduction

A Content-Addressable Memory (CAM) stores data in a similar fashion to a conventional RAM. However, "reading" the CAM involves providing input data to be matched, then searching the CAM for a match so that the address of the match can be output. Typical applications are networking, telecom (e.g., ATM cells), and consumer.

The CAM core in this application note is designed for Actel's ProASIC^PLUS and Axcelerator FPGA families. The Axcelerator multi-cycle architectures feature selectable parallelism, resulting from the variable-width ports available in those devices. Highlights include the following:

- Multiple architectures
  - Distributed and RAM-based
  - Multi-cycle and single-cycle read
- Implementation in Actel's ProASIC^PLUS and Axcelerator families
- VHDL code
- Device utilization
- Multi-cycle

  Axcelerator
  - 64x8 distributed RAM / AX125 / 1182 cells of 2016 (59%)
  - 1024x8 Block RAM (no parallelism) / AX500 /165 of 8064 cells (2%) + 2 1024x4 block RAMs
  - AX500 / 535 of 8064 cells (7%) + 8 32x32 block RAMs

  ProASIC^PLUS
  - 256x8 distributed RAM / APA600 / 8400 tiles of 21504 (39%)
  - 3072 tiles (10%) + 1 32x8 block RAM
- Single-cycle

  Axcelerator
  - 256x8 distributed RAM / AX500 / 5138 cells of 8064 (64%)

  ProASIC^PLUS
  - 32x8 distributed RAM / APA075 / 1037 tiles of 1072 (34%)

CAM Architecture

Writing to a CAM is exactly like writing to a conventional RAM. However, the “read” operation is actually a search of the CAM for a match to an input "tag." In addition to storage cells, the CAM requires one or more comparators (see Figure 1 on page 2 for a single CAM cell). The Actel cores described below use different numbers of comparators depending on the performance requirements. Architectures can also include an input address bus. Another common scheme involves writing to consecutive locations of the CAM as new data is added. The outputs are a MATCH signal (along with an associated MATCH_VALID signal) and either an encoded N-bit value or a one-hot-encoded bus with one match bit corresponding to each CAM cell. Both of these architectures are provided in the Axcelerator and ProASIC^PLUS FPGA families.

The multi-cycle CAM architecture tries to find a match to the input data word by simply sequencing through all memory locations – reading the contents of each location, comparing the contents to the input value, and stopping when a match is found (Figure 2 on page 2). At that point, MATCH and MATCH_VALID are asserted. If no match is found, MATCH is not asserted, but MATCH_VALID is asserted after all addresses are compared. MATCH_VALID indicates the end of the read cycle. In other words, MATCH_VALID asserted and MATCH not asserted indicate that all the addresses have been compared during a read operation and no matches were found.

When a match is found, the address of the matching data is provided as an output and the MATCH signal is asserted. It is possible that multiple locations might contain matching data, but no checking is done for this. Storage for the multi-cycle CAM can be either in distributed RAM (registers) or block RAM. Axcelerator RAM blocks have a useful size of 4,096 bits for this application, while the corresponding size for ProASICPLUS is 2,048 bits.

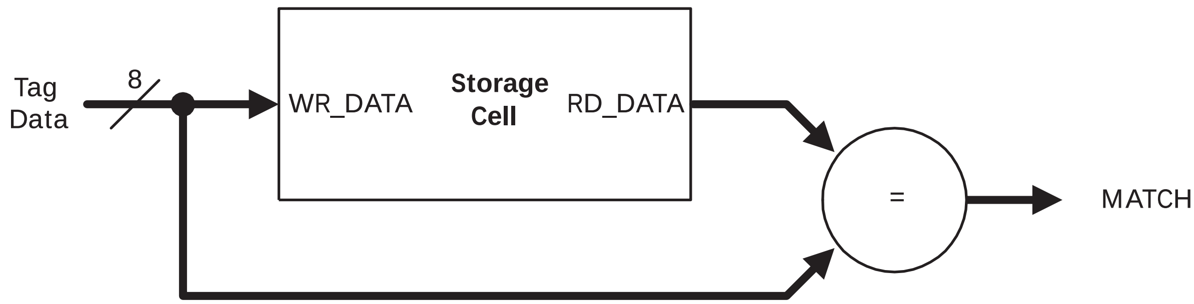

Figure 1 • **CAM Cell Block Diagram**

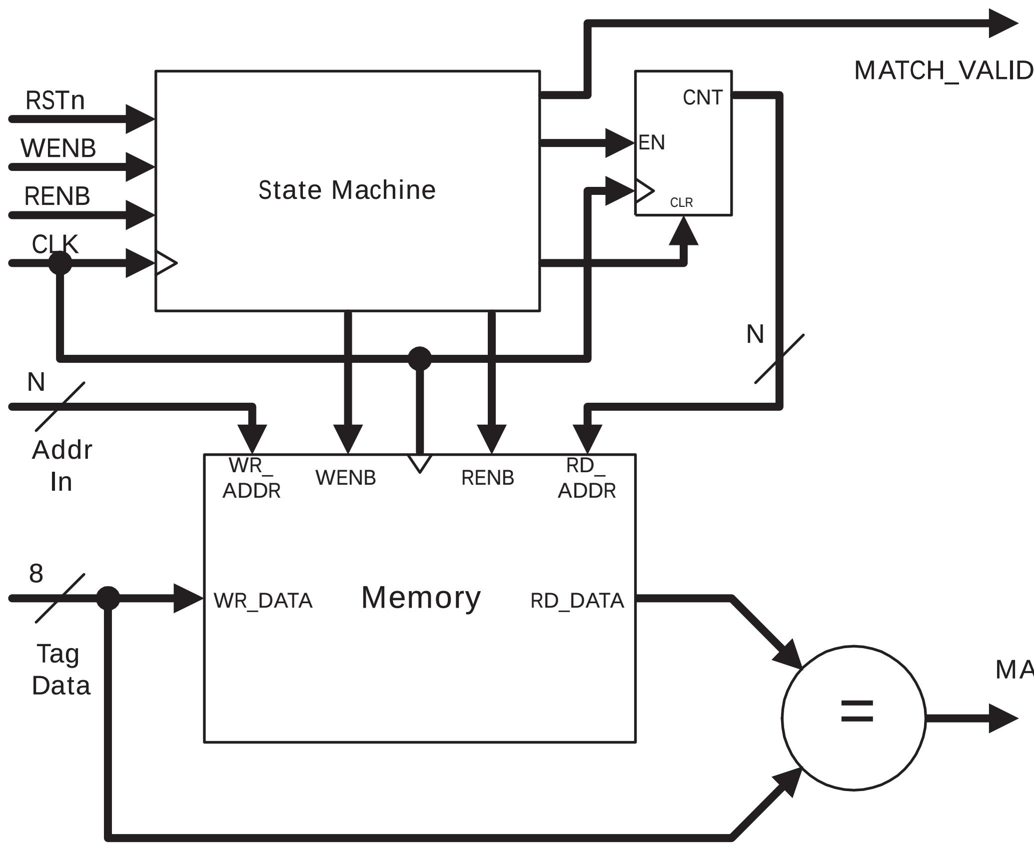

Figure 2 • **Multi-Cycle CAM Architecture**

The single-cycle CAM architecture is shown in Figure 3. In this case, there is a comparator for each storage location. If there are N CAM locations, the output is a single N-bit MATCH signal, which represents the one-hot encoding of the comparator results at each location. A '1' indicates a match and a '0' represents no match. Note that a match at any location triggers assertion of the MATCH signal; no checking is done vis-à-vis multiple addresses, which contain matching data. The storage cells are implemented as distributed registers. For the Actel single-cycle CAM generators, you can choose to have a single MATCH output, which is the logical OR of all of the MATCH bits

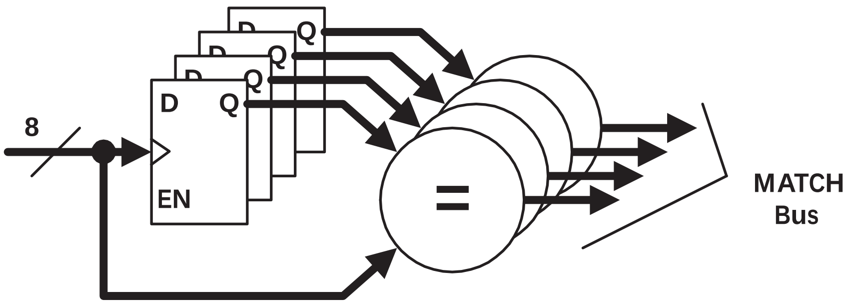

Figure 3 • **Single-Cycle CAM Architecture**

Increasing Parallelism in Multi-Cycle CAMs

The read time for a simple multi-cycle CAM is proportional to the number of locations. This time can be reduced substantially by using multiple memories in parallel and increasing the number of comparators. The Actel generators accommodate two stages of parallelism. Stage 1 uses multiple memories in parallel and a comparator for each memory block. This approach can be used in either ProASIC<u>PLUS</u> or Axcelerator FPGAs. A generic in a VHDL package file allows setting the amount of parallelism to the integer value K, where 2^K is the number of memories. If the initial number of address locations is N, then a single Nx8 memory with a single comparator will be replaced by 2^K $(N/2^K)$x8 memories, each with its own comparator. The default is K=0, corresponding to a single memory and comparator. K can also be set to 1, 2, or 3. The Stage 1 approach is illustrated in Figure 4.

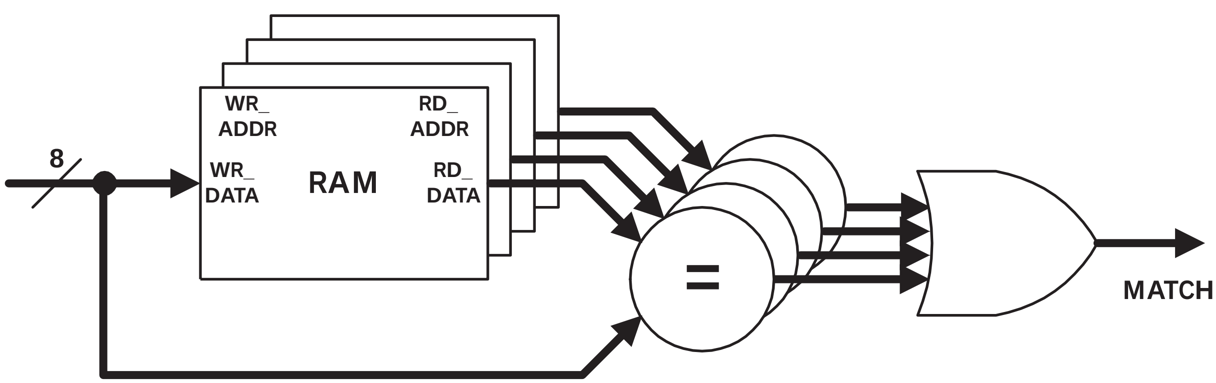

Figure 4 • **Multi-Cycle CAM Parallelism – Stage 1**

Stage 2 is only available in the Axcelerator family. It takes advantage of the built-in bus-width conversion feature of the embedded memory blocks in Axcelerator. The write port is eight bits, but the read port is 32 bits and uses four comparators per port as shown in Figure 5. This method reduces the number of clock cycles required to scan the memory block by effectively reading four eight-bit locations simultaneously and performing four comparisons during each clock cycle. The 4:1 parallelism, combined with the maximum parallelism of 8:1 in Stage 1, gives an approximate 32X performance increase vis-a-vis the 'standard' multi- cycle CAM implementation.

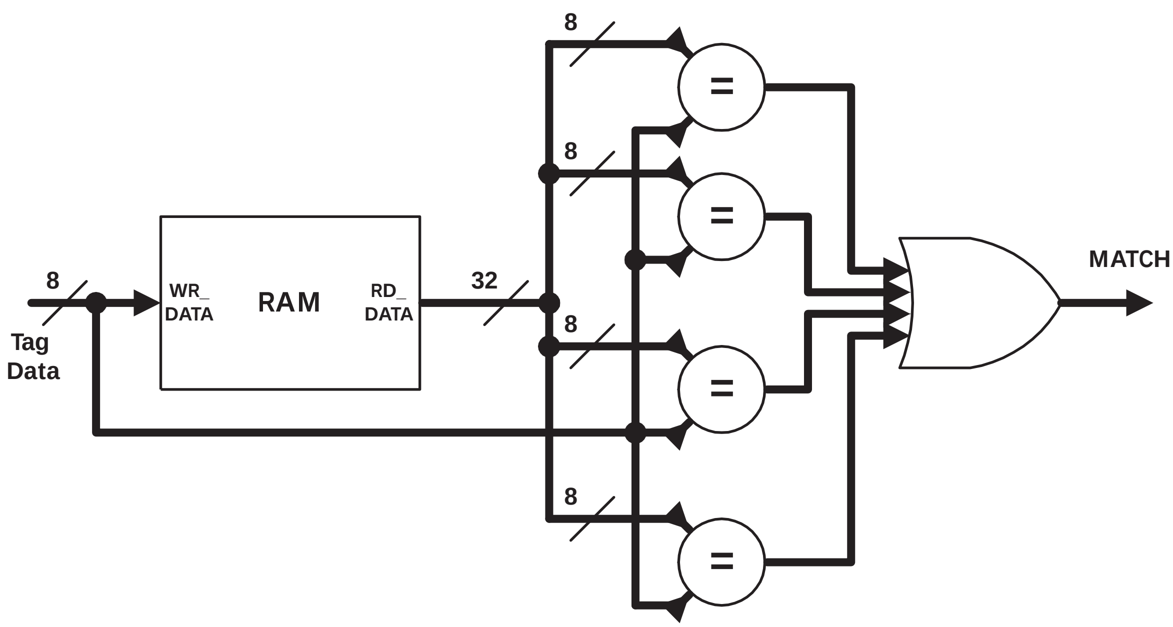

Figure 5 • **Multi-Cycle Parallelism – Stage 2 – Variable-Port RAM in Axcelerator**

Signal Descriptions

Signal	Function	Description
CLK	Input	Clock
RESETn	Input	CAM Controller Reset (Active Low). This signal does NOT clear the stored values in the CAM
WRB	Input	CAM Write Enable (Active Low). This signal, when driven low, enables writing of the eight-bit Tag Data bus to the RAM location specified by ADDR_IN on the rising edge of CLK
RDB	Input	SDRAM Read Request. This signal, when driven low for a minimum of one cycle, enables the search of the CAM for data, which matches that on the TAG_DATA bus. For the multi-cycle CAM, a search continues until a match is found or the entire CAM has been addressed. The single-cycle CAM search is completed two cycles after the assertion of RDB and a corresponding rising edge of the CLK signal
ADDR_IN(N-1:0)	Input	N-bit Input Address bus
TAG_DATA(7:0)	Input	Input data for writing or matching (see WRB and RDB above)
ADDR_OUT(N-1:0)	Output	*Multi-cycle CAM Only* – N-bit bus indicating the encoded address of matching data following a CAM read cycle. A valid match is indicated by assertion of MATCH. However, if MATCH is NOT asserted, the read cycle may not be complete; see the description of MATCH_VALID below. *Single-cycle CAM Only* – N-bit address bus indicating the one-hot address of matching data following a CAM read cycle
MATCH	Output	When asserted, indicates a MATCH has been found during the current read cycle. This signal is deasserted at the beginning of a read cycle. *Multi-cycle CAM Only* – When deasserted, no match may be present OR read cycle may not be complete. See MATCH_VALID below
MATCH_VALID	Output	When asserted, indicates that a CAM read cycle is complete and that the match indication from the MATCH signal is valid

Implementation and Performance/Utilization Summaries

The implementation is mostly behavioral, but key blocks are implemented structurally to target features of Actel devices. Both architectures are implemented in VHDL. Generics allow setting the number of address bits, the memory depth and width, and the degree of parallelism. The synthesis results given below were from Synplicity version 7.2.1. Simulation was done using Model*Sim* 5.5e.

Simulation results for the multi-cycle CAM are shown in Figure 6 on page 6. To make the simulation easier to follow, the data written to a given address is the address itself. Four writes occur (note that WRB is low for four cycles). The signals prefixed with 'disp' are internal signals. Note that for a CAM read, RDB needs only be asserted for one cycle. The data to be matched is '00000010' in this simulation. The internal memory read enable, disp_renb', is then asserted continuously until either a match is found (as in this case) or until all memory addresses have been read and checked. When the match is found, ADDR_OUT is set to '00000010' and MATCH and MATCH_VALID are asserted.

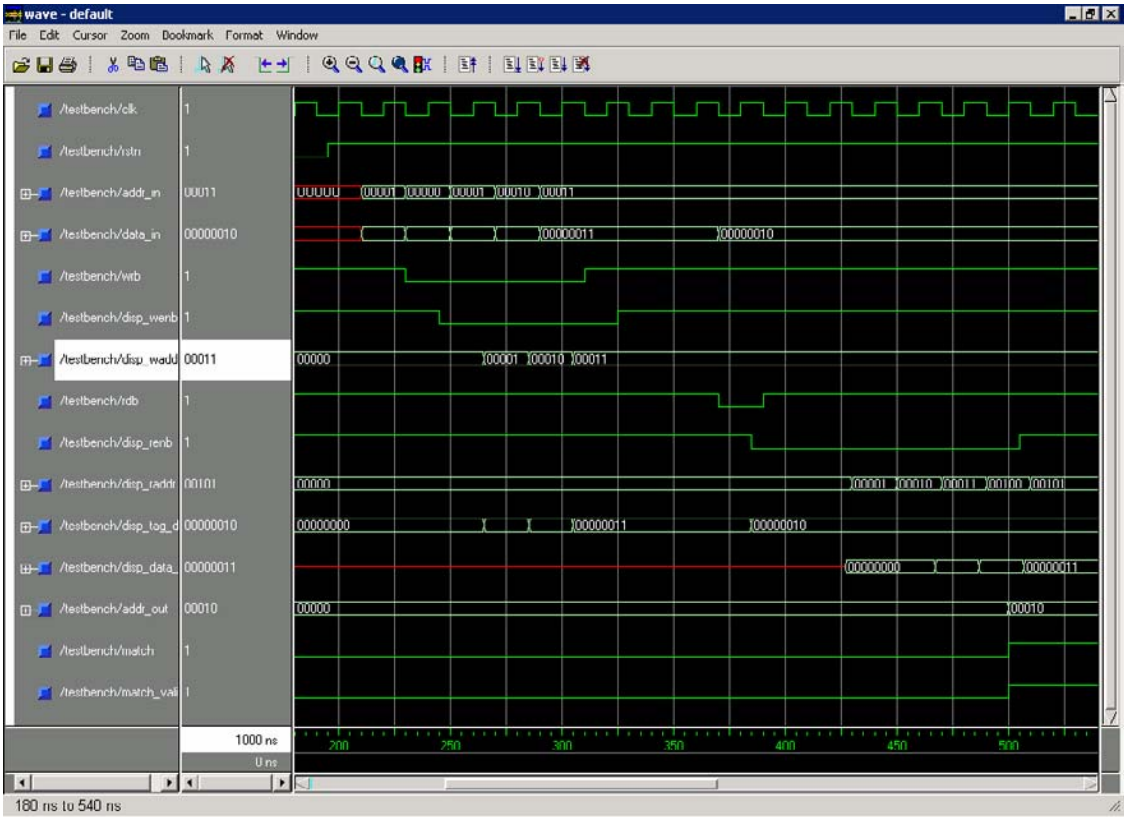

Figure 6 • **Simulation Results**

The performance and utilization statistics are given in Table 1 through Table 5 on page 7 for various CAM architectures. Timing numbers are given for either the Axcelerator –3 speed grade or the ProASIC<u>PLUS</u> Std. speed grade.

Table 1 • **Axcelerator Multi-Cycle CAM Utilization and Performance Statistics (no parallelism)**

	Utilization (Estimated)					Performance	
	Comb. Modules	Seq. Modules	Total Modules	% (Device)	RAM Blocks	Max. Freq (MHz)	Min. Cycle (ns)
32x8 Distributed	314	280	594	30 (AX125)	0	120	8.3 (Write) 274 (Read)
64x8 Distributed	655	527	1182	59 (AX125)	0	101	9.9 (Write) 643 (Read)
32x8 with Block RAM	103	59	162	2 (AX500)	1	144	6.9 (Write) 241 (Read)
1024x8 with Block RAM	99	66	165	2 (AX500)	2	110	9.1 (Write) 10,500 (Read)

Table 2 • **Axcelerator Multi-Cycle CAM Utilization and Performance Statistics (1024x8 Block RAM with and without Parallelism)**

	Utilization (Estimated)					Performance	
	Comb. Modules	Seq. Modules	Total. Modules	% (Device)	RAM Blocks	Max. Freq (MHz)	Min. Cycle (ns)
No Parallelism	99	66	165	2 (AX500)	2	110	9.1 (Write) 10,500 (Read)
Stage 1	193	81	274	3 (AX500)	8	99	10.1 (Write) 1,300 (Read)
Stage 2	434	101	535	7 (AX500)	8	88	11.4 (Write) 399 (Read)

Table 3 • **ProASIC^PLUS Multi-Cycle CAM Utilization and Performance Statistics (No Parallelism)**

	Utilization			Performance	
	Tile Count	% (Device)	RAM Blocks	Max. Freq (MHz)	Min. Cycle (ns)
16x8 Distributed	1042	34 (APA075)	0	49	20.4 (Write) 347 (Read)
32x8 Distributed	2039	66 (APA075)	0	38	26.3 (Write) 868 (Read)
256x8 Distributed	8400	39 (APA600)	0	26	38.5 (Write) 9,900 (Read)
32x8 w/ Block RAM	300	10 (APA075)	1	28	35.7 (Write) 1,250 (Read)

Table 4 • **Axcelerator Single-Cycle CAM Utilization and Performance Statistics**

	Utilization (Estimated)				Performance	
	Comb. Modules	Seq. Modules	Total Modules	% (Device)	Max. Freq (MHz)	Min. Cycle (ns)
64x8 Distributed	732	576	1308	65 (AX125)	175	5.7 (Clk-to-Match)
256x8 Distributed	2834	2304	5138	64 (AX500)	172	5.8 (Clk-to-Match)

Table 5 • **ProASIC^PLUS Single-Cycle CAM Utilization and Performance Statistics**

	Utilization		Performance	
	Tile Count	% (Device)	Max. Freq (MHz)	Min. Cycle (ns)
32x8 Distributed	1037	34 (APA075)	172	5.8 (Clk-to-Match)

Conclusion

A variety of VHDL CAM generators are available, which are targeted to the Actel ProASIC^PLUS and Axcelerator FPGA families. CAM storage is done either with distributed registers or with block RAM available in these families. Performance can be enhanced by using multiple comparators; the degree of parallelism can be chosen by the user.

http://www.actel.com

Actel Corporation
2061 Stierlin Court
Mountain View, CA
94043-4655 USA
Tel: (650) 318-4200
Fax: (650) 318-4600

Actel Europe Ltd.
Dunlop House, Riverside Way
Camberley, Surrey GU15 3YL
United Kingdom
Tel: +44 (0)1276 401450
Fax: +44 (0)1276 401490

Actel Japan
EXOS Ebisu Bldg. 4F
1-24-14 Ebisu Shibuya-ku
Tokyo 150 Japan
Tel: +81 03-3445-7671
Fax: +81 03-3445-7668

Actel Hong Kong
39th Floor
One Pacific Place
88 Queensway
Admiralty, Hong Kong
Tel: 852-22735712

51900065-0/12.03