Chapter 2
An Overview of Content Addressable Memory

In this chapter, it introduces the overview of CAM. The conventional CAM architecture, CAM cell circuits, and CAM word schemes would be described in Section 2.1, Section 2.2, and Section 2.3, respectively. In addition, the applications of CAM would be detailed in Section 2.4. Finally, Section 2.5 would give an introduction for low power content-addressable memory design.

2.1 Conventional CAM Architecture

A conventional CAM architecture is usually composed of the data memories, address decoders, bit-lines pre-charge circuits, word-line match schemes, read sense amplifiers, address priority encoders and so on [19-21]. Fig.2.1 shows a simplified block diagram of a CAM. Generally, CAM has three operation modes: write, read, and search. In write and read operation, CAM plays just like an ordinary memory. Different from SRAM, CAM has a search mode. The input in Fig. 2.1 called search word is broadcasting onto the search-lines to the table of stored data. The number of bits in a CAM word is usually large, with existing implementations ranging from 36 to 144 bits. A typical CAM employs a table size ranging from a few hundred entries to 32K entries, corresponding to an address space ranging from 7 bits to 15 bits. Each stored word has a match-line that indicates whether the search word and the stored word are identical (the match case) or different (a mismatch case). The match-lines are fed to an encoder that generates a binary match location corresponding to the match-line that is in the match state. An encoder is used in systems where only a single match is expected. If more than one word may match, a priority encoder is used to select the highest priority matching location to map to the match result. The overall function of a CAM is to take a search word and return the matching memory location.

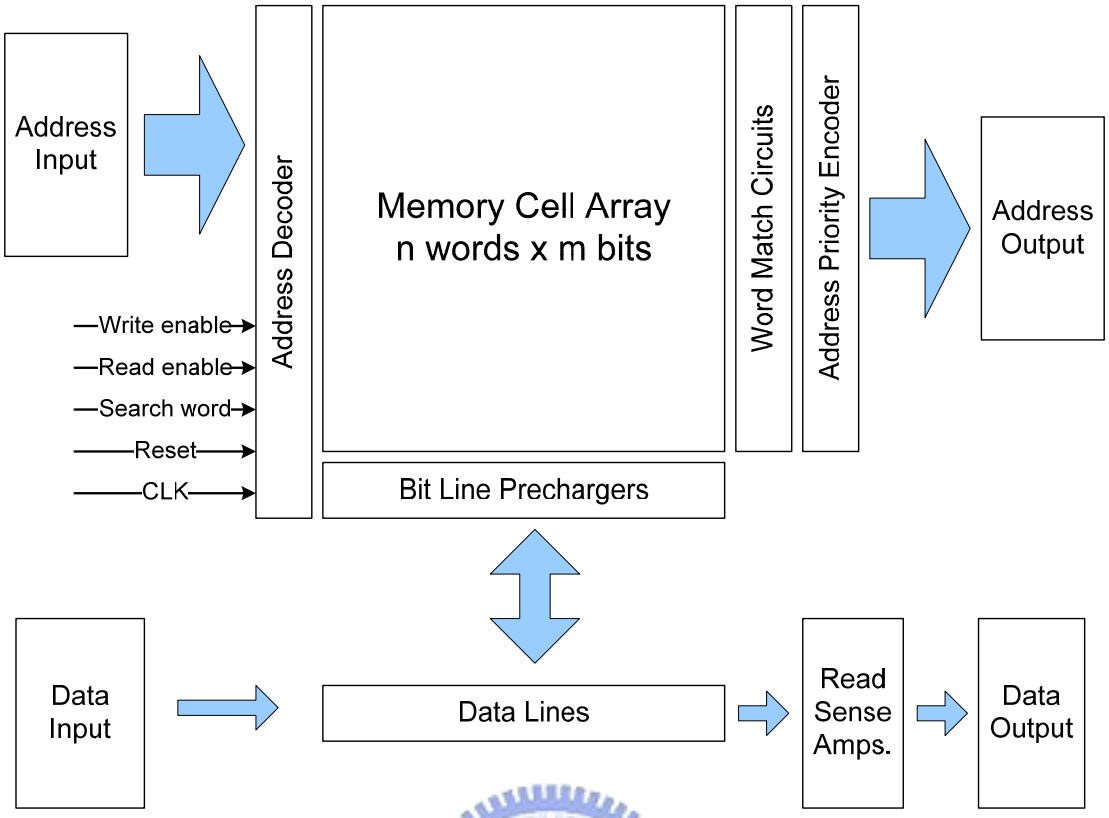

Fig. 2.1 Conventional CAM Architecture

2.2 Conventional CAM Cell

In a conventional CAM cell, there are two types of CAM cells would: one is binary CAM (BCAM) cell and the other is ternary CAM (TCAM) cell.

2.2.1 Binary CAM Cell

Depending on the different searching operation, CAM[1] cells are classified into NOR-type CAM cell and AND-type[2] CAM cell [22]. Fig. 2.2 depicts the NOR-type CAM cells which are widely used in past years. Fig. 2.2 (a) is constructed by 9-transistor structure and Fig. 2.2 (b) is composed of 10-transistor structure. Table 2.1 shows the truth table of a NOR-type CAM cell. The 9-transistor CAM cell consists of

[1] In this thesis, we call binary CAM as CAM simply. If we mean the ternary CAM, we will call TCAM specially.

[2] Because the principles and operations of AND-type cells like ones of NAND-type cells, we all call NAND-type match-line scheme as AND-type match-line scheme in this thesis.

a traditional 6-transistor SRAM and a PTL-type compare circuit; the 10-transistor CAM cell is composed of an ordinary 6-transistor SRAM and the pull down XOR comparison circuits. In read and write operation, both 9-transistor and 10-transistor CAM cell work the same as a SRAM cell. In the search mode, as for the 9-transistor cell, the match-line will be charged to high first. If the search data is equal to the stored data, the node X becomes low, turning Mn off and leaving match-line still floating high. On the other hand, if the search data doesn't match with the stored data, the node X would become high and Mn being turned on. Therefore, the match-line would be discharged to ground. Regarding 10-transistor CAM cells, the principle is same as 9-transistor CAM cells. During search operation, the match-line would be pre-charged to high first. If searching data is equal to the stored data, the match-line is still floating. Contrarily, if searching data is not equal to the stored data, there is a path from match-line to ground and match-line would be discharged to ground.

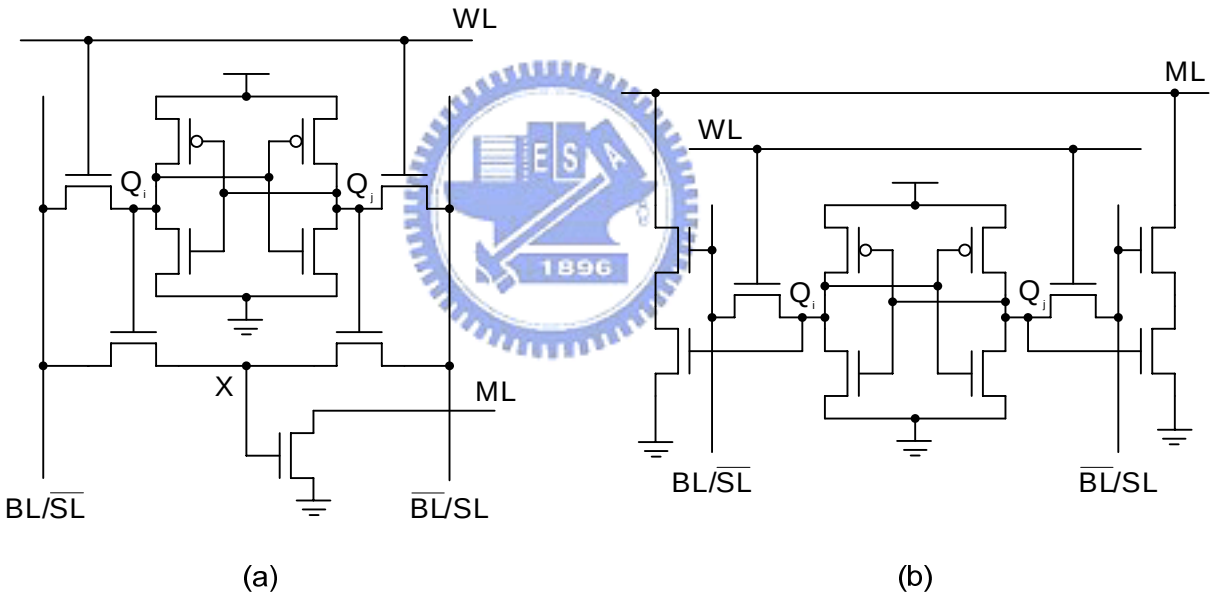

Fig. 2.2 NOR-type binary CAM cell. (a) 9-transistor cell and (b) 10-transistor cell.

Table 2.1 Truth table of NOR-type binary CAM cell.

State	**Q_i**	**SL**	**ML**
Zero (0)	0	0	floating
	0	1	0
One (1)	1	0	0
	1	1	floating

An AND-type CAM cell is similar to 9-transistor CAM cell whatever it works in write or read operation. The only difference from 9-transistor CAM cell is the match-line scheme. Fig. 2.3 depicts an AND-type CAM cell and Table 2.2 describes the truth table of AND-type CAM cell. As an AND-type CAM cell works in search operation, the match-line would be pre-charged to high first. However, contrary to 9-tansistor NOR-type CAM cell, the match-line hold floating when the search data doesn't match with stored data and the match-line is discharged to ground only while the search data and stored data are match.

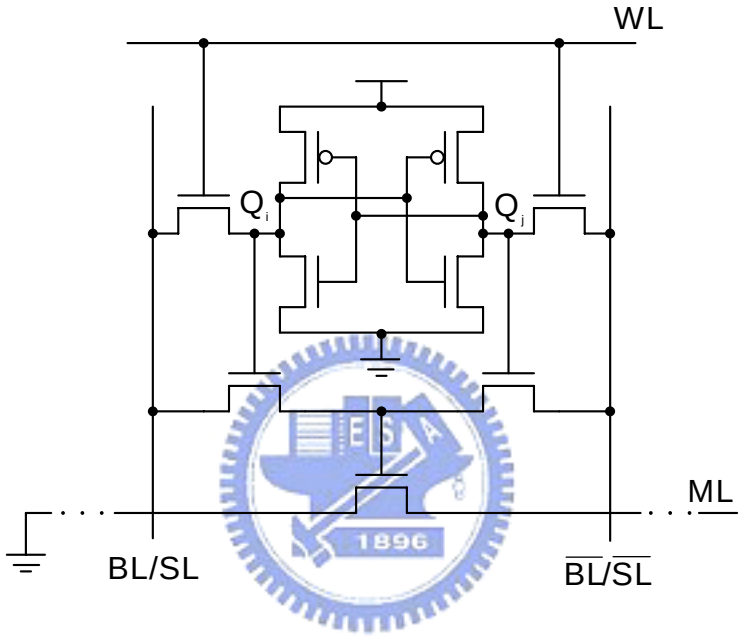

Fig. 2.3 AND-type 9-transistor binary CAM cell.

Table 2.2 Truth table of AND-type binary CAM cell.

State	Q_i	SL	ML
Zero (0)	0	0	0
	0	1	floating
One (1)	1	0	floating
	1	1	0

2.2.2 Ternary CAM Cells

For the CAM circuit design, the ternary CAM (TCAM) performs a more powerful data search function [23-24]. Different from binary CAM which has two

states: one (1) and zero (0) state, the ternary CAM (TCAM) cell has an additional state: don't care (X) state. TCAM is also classified into two kinds: NOR-type TCAM cell and AND-type TCAM cell.

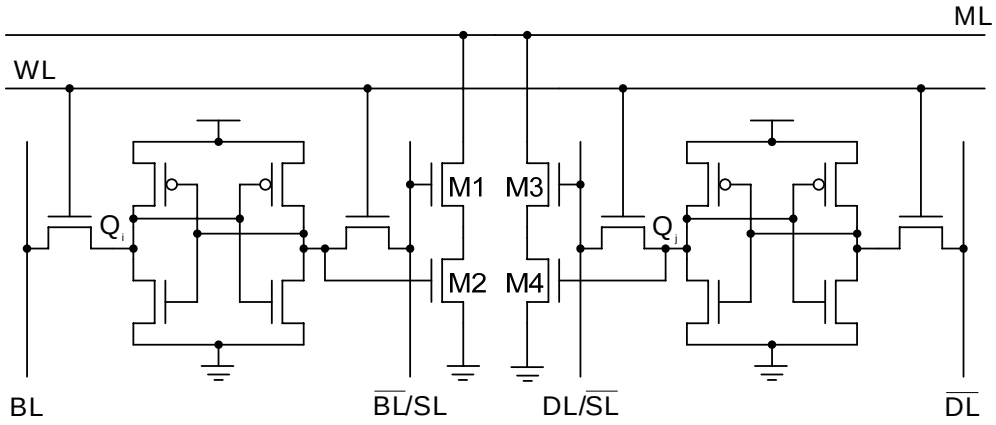

Fig. 2.4 Static NOR-type ternary CAM cell.

Table 2.3 State assignments and truth table for static TCAM cell.

State	Q_i	Q_j	SL	ML
Zero (0)	0	1	0	floating
	0	1	1	0
One (1)	1	0	0	0
	1	0	1	floating
Don't care (X)	0	0	0	floating
	0	0	1	floating
Not allowed	1	1	0	—
	1	1	1	—

Fig. 2.4 shows a static NOR-type TCAM cell. It consists of 2-SRAM and 4-transistor comparison circuits. This TCAM cell is designed to store three states, namely zero (0), one (1) and don't care (X). These three states are set by Q_i and Q_j. Table 2.3 illustrates the truth table of the three state of the static NOR-type TCAM cell. When Q_i is low and Q_j is high, the TCAM cell is in the "zero" state. In the searching operation, the same as BCAM cell, match-line will be charged to high first. If search data is low, the NMOS M1 and M4 would not be turned on, such that the ML will still be floating. On the other hand, while search data is high, the NMOS M1 and M2 are turned on at the

same time result in the match-line being discharged to ground. However, the TCAM cell is in the "one" state, while search data is high, the match-line would keep high. While search data is low, the match-line would be discharge to the ground. Particularly, while Q_i and Q_j are both low, the TCAM cell is in "don't care" state. No matter search data is high or is low, the NMOS M1 and M3 are not turned on result in the match-line keeping floating. Note that Q_i and Q_j cannot be high simultaneously, this state are not be allowed.

There is an additional NOR-type TCAM cell called dynamic TCAM cell [25-29], as shown in Fig. 2.5. The major difference between static TCAM cell and dynamic TCAM cell is that the storage cells composed of 2 SRAM cells in static TCAM cell are replaced by 2 capacitances in dynamic TCAM cell. The dynamic TCAM cell works like static TCAM and the truth table of static TCAM shown in Table 2.3 also applies to dynamic TCAM.

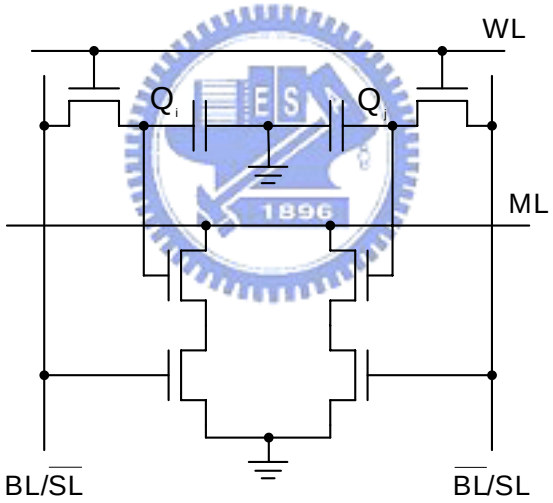

Fig. 2.5 Dynamic NOR-type ternary CAM cell.

Fig. 2.6 illustrates a 16-transistor AND-type TCAM cell which includes 2-SRAM and comparison circuits composed of three NMOS. The state assignments and truth table of this TCAM cell is described in Table 2.4. The AND-type TCAM cell is alike a 9-transistor AND-type BCAM cell when TCAM cell works in zero (0) and one (1) states. However, while this AND-type TCAM cell is in don't care (X) state (Q_j is high), no matter the search data is high or low, the match-line would be discharged.

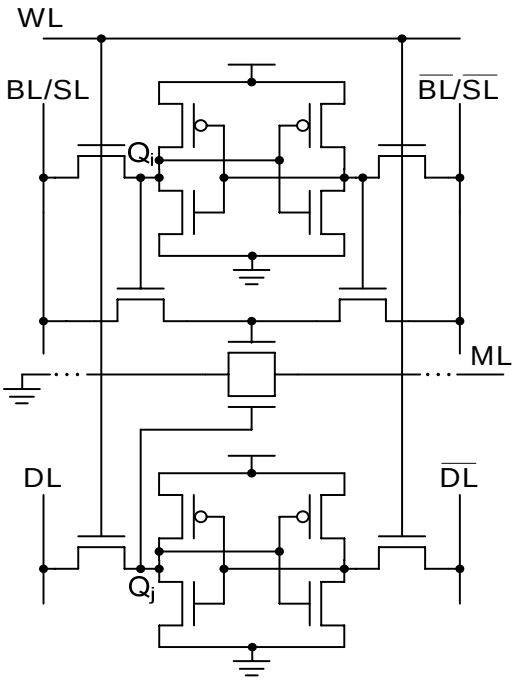

Fig. 2.6 AND-type ternary CAM cell.

Table 2.4 State assignments for TCAM cell.

State	Q_i	Q_j	SL	ML
Zero (0)	0	0	0	0
	0	0	1	floating
One (1)	1	0	0	floating
	1	0	1	0
Don't Care (X)	0	1	0	0
	0	1	1	0
	1	1	0	0
	1	1	1	0

2.3 Match-line Structure

In the conventional CAM architecture, the CAM word circuits adopts dynamic CMOS circuits to improve data matching performance and hardware cost. the conventional NOR-type CAM word schemes and AND-type match-line schemes are shown in Fig. 2.7 (a) and Fig. 2.7 (b), respectively [30].

2.3.1 NOR-type Match-line

Fig. 2.7 (a) depicts, in schematic form, how NOR cells are connected in parallel to form a NOR match-line, ML. While we show binary cells in the figure, the description of match-line operation applies to both binary and ternary CAM. A typical NOR search cycle operates in three phases: search-line precharge, match-line precharge, and match-line evaluation. First, the search-lines are precharged low to disconnect the match-lines from ground by disabling the pull down paths in each CAM cell. Second, with the pull down paths disconnected, the M_{pre} transistor precharges the match-lines high. Finally, the search-lines are driven to the search word values, triggering the match-line evaluation phase. In the case of a match, the ML voltage, V_{ML} stays high as there is no discharge path to ground. In the case of a miss, there is at least one path to ground that discharges the match-line. The match-line sense amplifier (MLSA) senses the voltage on ML, and generates a corresponding full-rail output match result. We will see several variations of this scheme for evaluating the state of NOR match-lines in Section III. The main feature of the NOR match-line is its high speed of operation. In the slowest case of a one-bit miss in a word, the critical evaluation path is through the two series transistors in the cell that form the pull down path. Even in this worst case, NOR-cell evaluation is faster than the NAND case, where between 8 and 16 transistors form the evaluation path.

2.3.2 AND-type Match-line

Fig. 2.7 (b) shows the structure of the AND match-line. A number of AND CAM cells are cascaded to form the ML (this is, in fact, a floating node, but for consistency we will refer to it as ML). For the purpose of explanation, we use the binary version of the AND cell, but the same description applies to a ternary cell. On the right of the figure, the precharge pMOS transistor, M_{pre} sets the initial voltage of the ML to the supply voltage, and then the evaluation nMOS transistor, $N_{p,}$ turns ON. In the case of a match, all nMOS transistors are ON, effectively creating a path to ground from the ML, hence discharging ML to ground. In the case of a miss match, at least one of the series nMOS transistors is OFF, leaving the ML voltage high. The AND match-line has an explicit evaluation transistor, N_p, unlike the NOR match-line, where the CAM

cells themselves perform the evaluation.

There is a potential charge-sharing problem that can occur between the ML and the intermediate nodes. in the AND type match-line. For example, in the case where all bits match except for the leftmost bit in Fig. 2.7 (b), during evaluation there is charge sharing between the ML and nodes Nd_{nn-1} through Nd_{n1}. This charge sharing may cause the ML voltage to drop sufficiently low such that the output inverter detects a false match. A technique that eliminates charge sharing is to precharge high, the intermediate nodes. However, the power consumption increases. A feature of the AND match-line is that a miss stops signal propagation such that there is no power consumption past the final matching transistor in the serial nMOS chain. Typically, only one match-line is in the match state, consequently most match-lines have only a small number of transistors in the chain that are ON and thus only a small amount of power is consumed. Two drawbacks of the AND match-line are a quadratic delay dependence on the number of cells, and a low noise margin.

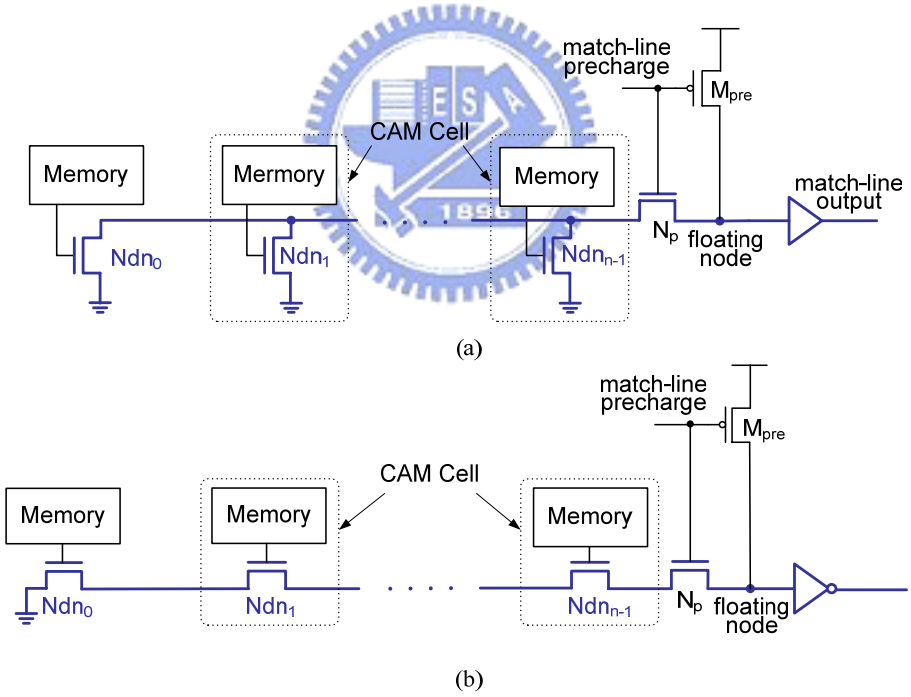

Fig. 2.7 Conventional CAM word circuits. (a) NOR-type and (b) AND-type

2.4 Applications of CAM

CAMs are widely used in cache memory system and translation look-aside buffer (TLB) in virtual memory system in past years. The primary commercial

application of CAMs today is to classify and forward Internet protocol (IP) packets in network routers [31-36]. A router, which is the intermediate node of the network, is used to compare the destination address of a packet to all possible routes, in order to choose the appropriate one. A CAM is a good choice for implementing this lookup operation due to its fast search capability.

2.4.1 Cache Memory

In the memory hierarchy system, cache plays an important role [37-38]. Cache is the first level of the memory hierarchy encountered once the address leaves the CPU. Its function is to refer to any storage managed to take advantage of locality of access. Cache serves as a method for providing fast reference to recently used portion of instruction or data. When CPU finds a wanted data item in the cache, it is called cache hit. On the contrary, if CPU does not find a data item that is needed in the cache, it is called cache miss. Temporal locality means that the requested data item is likely needs it again in the near future, so it is useful to place the requested data item in the cache where it can be accessed quickly. A fixed-size collection of data items which contains the requested data item is called block. There is high possibility that the other data items in the block will be needed soon for spatial locality.

An example for direct data mapping cache is illustrated in Fig. 2.8. The 32 bits long address is divided into three parts. First one is byte offset which occupies two bits. Second part is Index, and third part is Tag. The numbers of Index indicates the capacity of cache. An N-bits Index indicates 2^N entries in the cache which can be used to stored data items. The action is first to find the corresponding position of index. When the corresponding position is found out, the tag stored in the corresponding position would be taken out. This tag would be compared to the third part of address. If they are the same, and valid bit is one, a hit signals and the corresponding data would be sent out. Of course, the tag entries are composed of CAM array. The valid bit is used to indicate whether an entry contains a valid address. If they are not the same, a miss occurs, which means no requested data in the cache. The requested data may be stored in the lower level memory. When the requested data is found in the lower level, it would be written back to the cache.

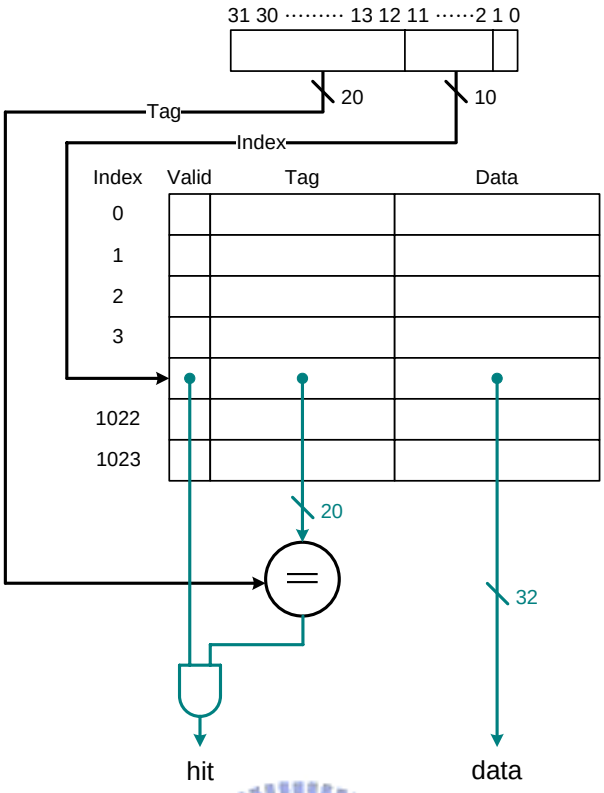

Fig. 2.8 A simple cache memory.

2.4.2 Translation Look-aside Buffer

Translation look-aside buffer (TLB) widely used in virtual memory system can speed up address translation in processor with virtual memory and can also cut down access time and lowering the miss rates. A TLB is like a cache that hold only page table mapping [37-38]. It provides fast translation from the virtual address to the physical address. When we get the physical address, we can use this physical address to access the data stored in the physical memory, such as cache, DRAM or DISK.

Fig. 2.9 shows a simple virtual memory system. The TLB contains a subset of the virtual-to-physical page mappings that are in the page table. Because the TLB is a cache, it must have a tag field which consists of CAMs. As a virtual page number (VPN) is sent to the TLB, this VPN would be compared with all valid tags in TLB. If a corresponding tag in the TLB is found, the corresponding physical page address can be found. If there is no matching entry in the TLB instead, the page table must be examined. The page table either supplies a physical page number for the page or indicates that the page resides on disk, in which case a page fault occurs.

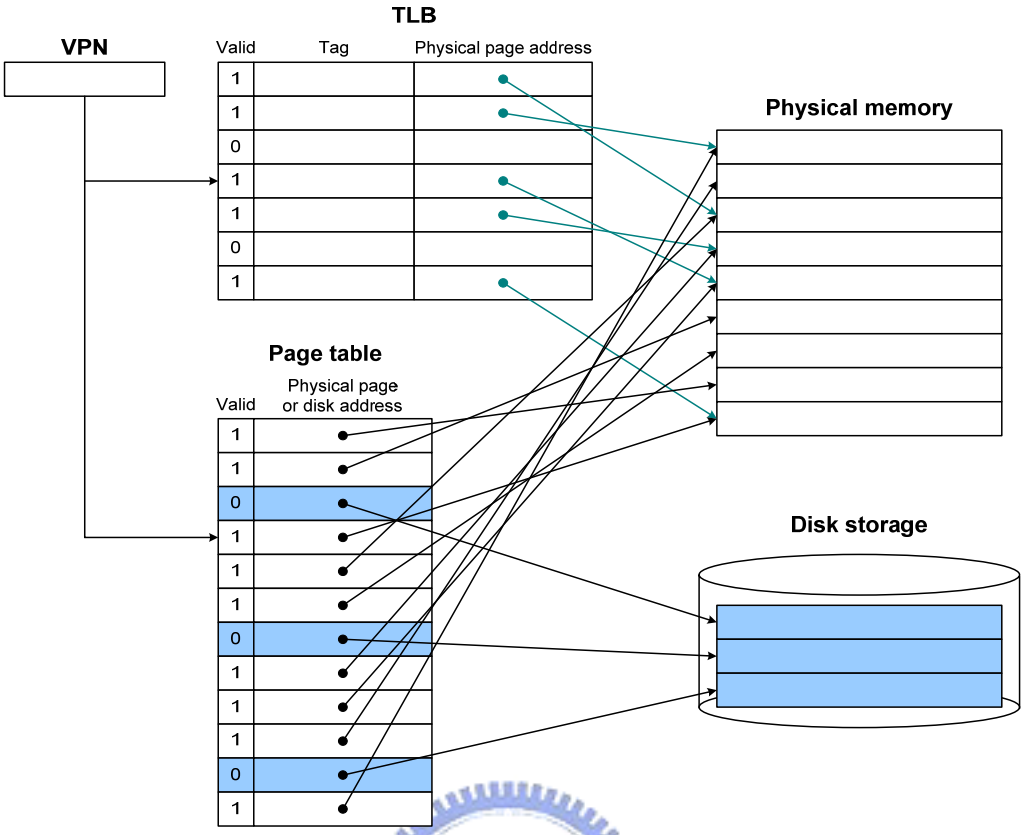

Fig. 2.9 A simple virtual memory system.

2.4.3 Packet Forwarding Using CAM

In networks like the Internet, a message such as an e-mail or a Web page is transferred by first breaking up the message into small data packets of a few hundred bytes, and then sending each data packet individually through the network. These packets are routed from the source, through routers, and reassembled at the destination to reproduce the original message.

Further examples of approaches using CAMs for sorting and searching are provided in [39-40]. Network routers forward data packets from an incoming port to an outgoing port, using an address-lookup function. This function examines the destination address of the packet and selects the output port associated with that address. The router maintains a list, called the routing table, which contains destination addresses and their corresponding output ports. An example of a simplified routing table is displayed in Table 2.5. All four entries in the table are 5-bit words with the don't care bit, "X". Because of the "X" bits, the first three entries in

the Table represent a range of input addresses, i.e., entry 1 maps all addresses in the range 10100 to 10111 to port A. The router searches this table for the destination address of each incoming packet, and selects the appropriate output port. For example, if the router receives a packet with the destination address 10100, the packet is forwarded to port A. In the case of the incoming address 01101, the address lookup matches both entry 2 and entry 3 in the table. Entry 2 is selected since it has the fewest "X" bits, or, alternatively, it has the longest prefix, indicating that it is the most direct route to the destination. This lookup method is called longest-prefix matching. Fig. 2.10 illustrates how a CAM accomplishes address lookup by implementing the routing table shown in Table 2.5. On the left of Fig. 2.10, the packet destination-address of 01101 is the input to the CAM. As in the table, two locations match, with the (priority) encoder choosing the upper entry and generating the match location 01, which corresponds to the most-direct route. This match location is the input address to a RAM that contains a list of output ports. A RAM read operation outputs the port designation; port B, to which the incoming packet is forwarded. We can view the match location output of the CAM as a pointer that retrieves the associated word from the RAM. This CAM/RAM system is a complete implementation of an address-lookup engine for packet forwarding.

Table 2.5 Example Routing Table

Entry NO.	**Address (Binary)**	**Output Port**
1	101XX	A
2	0110X	B
3	011XX	C
4	10011	D

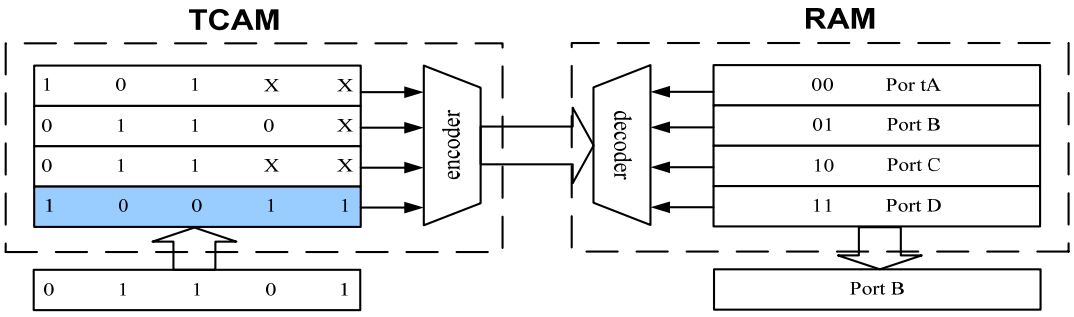

Fig. 2.10 CAM-based implementation of the routing table of Table 2.5

2.4.4 ATM Switches

For ATM switching network application, CAM can be adopted as a translation table. Virtual circuits are important parts to ATM networks, which need to be set up across ATM networks before any data transfer because ATM networks are connection-oriented. There are two types of ATM virtual circuits, Virtual Path (identified by a virtual path identifier [VPI]) and Channel Path (identified by a channel path identifier [VCI]). Each segment of the total connection has unique VPI/VCI combinations, and the VPI/VCI value of ATM cells would be changed into the value for the next segment of connection while ATM cell switches [41-42].

CAM is applied to an ATM switch as an address translator and can quickly perform the VPI/VCI translation. In the translation process, the CAM causes address which access data in RAM and uses incoming VPI/VCI values in ATM cell headers. A CAM/RAM combination realizes the multi-megabit translation tables with full parallel search capability. Take VPI/VCI fields from the TM cell header and the list of current connections stored in the CAM array for comparison, as a result, CAM originates and address which is used to access an external RAM where VPI/VCI mapping data and other connection information is stored. The ATM controller uses the VPI/VCI data from the RAM for modifying the cell header, and the cell is sent to the switch, depicted in Fig. 2.11.

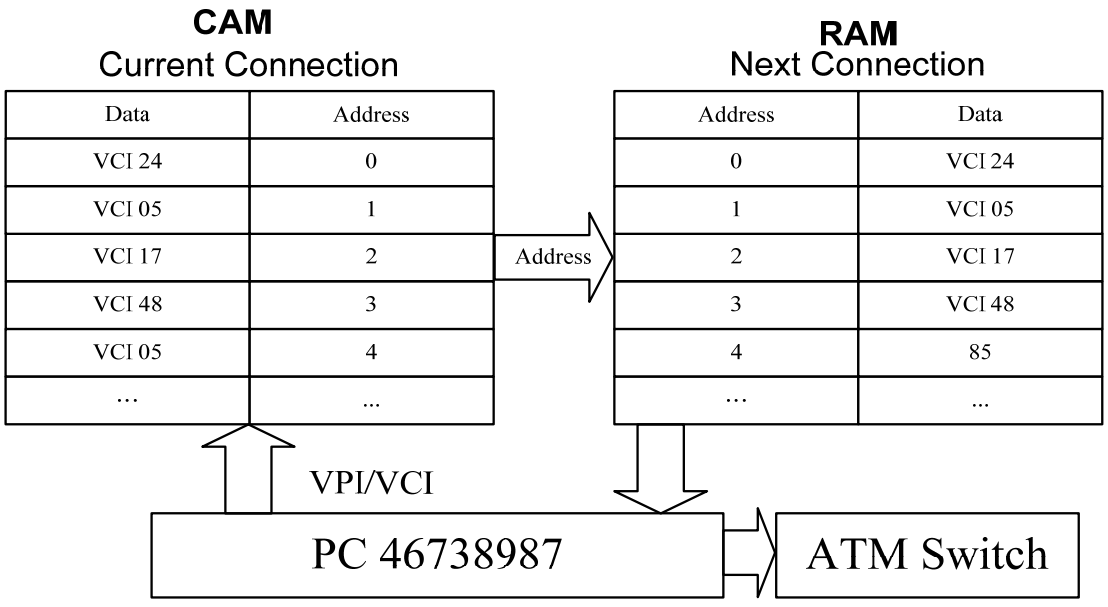

Fig. 2.11 ATM switch with CAM.

2.5 Low power Content-addressable Memory Designs

There are several low-power and high-speed CAMs were proposed [43-46]. In the past, many techniques were developed to improve the performance of CAM. On match-line scheme, there are low-swing scheme, current-race scheme, selective-precharge scheme, pipelining scheme and current-saving scheme. On search-line scheme, there are eliminating search-line precharge, charge recycling search-line driver and hierarchical search-line scheme. These designs will be introduced as follows.

2.5.1 Method of reducing Match-line power consumption

The dynamic power consumed by a single match-line that misses is due to the rising edge during precharge and the falling edge during evaluation, and is given by the equation

$$P_{miss} = C_{ML} V_{DD}^2 f \tag{2.1}$$

where f is the frequency of search operations. In the case of a match, the power consumption associated with a single match-line depends on the previous state of the match-line; however, since typically there is only a small number of match condition, we can neglect this power consumption. Accordingly, the overall match-line power consumption of a CAM block with ω match-lines is

$$P_{ML} = \omega P_{miss} = \omega C_{ML} V_{DD}^2 f \tag{2.2}$$

2.5.1.1 Low-Swing Scheme

One method of reducing the ML power consumption, and potentially increasing its speed, is to reduce the ML voltage swing [25], [27]. The reduction of power consumption is linearly proportional to the reduction of the voltage swing, resulting in the modified power equation

$$P_{ML} = \omega \times C_{ML} V_{DD} V_{MLswing} f \tag{2.3}$$

The match-line swing is reduced from V_{DD} to $V_{MLswing}$ where $V_{MLswing} < V_{DD}$. We can obvious that P_{ML} reduced.

2.5.1.2 Current-Race Scheme

Fig. 2.12 (a) shows a simplified schematic of the current-race scheme [48]. This scheme precharges the match-line low and evaluates the match-line state by charging the match-line with a current IML supplied by a current source. The signal timing is shown in Fig. 2.12 (b). The precharge signal, mlpre, starts the search cycle by precharging the match-line low. Since the match-line is precharged low, the scheme concurrently charges the search-lines to their search data values, eliminating the need for a separate SL precharge phase required by the precharge-high scheme. Instead, there is a single SL/ML precharge phase, as indicated in Fig. 2.12 (b). After the SL/ML precharge phase completes, the enable signal, $\overline{en}$, connects the current source to the match-line. A match-line in the match state charges linearly to a high voltage, while a match-line in the miss state charges to to a voltage of only $I_{ML} \times R_{ML} / m$, where m denotes the number of misses in cells connected to the match-line. By setting the maximum voltage of a miss to be small, a simple match-line sense amplifier easily differentiates between a match state and a miss state and generates the signal MLso. As shown in Fig. 2.12 (a), the amplifier is the nMOS transistor, M_{sense}, whose output is stored by a half-latch. The nMOS sense transistor trips the latch with a threshold of V_{tn}. After some delay, match-lines in the match state will charge to slightly above V_{tn} tripping their latch, whereas match-lines in the miss state will remain at a much smaller voltage, leaving their latch in the initial state. A simple replica match-line (not shown) controls the shutoff of the current source and the latching of the match signal.

The power consumption can be derived by first noting that the same amount of current is discharged into every match-line, regardless of the state of the match-line. Looking at the match case for convenience, the power needed to charge a match-line to slightly above V_{tn} is

$$P_{match} = C_{ML} V_{DD} V_{tn} f \tag{2.4}$$

Since the power consumption of a match and a miss are identical, the overall power consumption for all ω match-lines is

$$P_{ML} = \omega \times C_{ML} V_{DD} V_{tn} f \tag{2.5}$$

This equation is identical to the low-swing scheme (2.3), with $V_{MLswing} = V_{tn}$

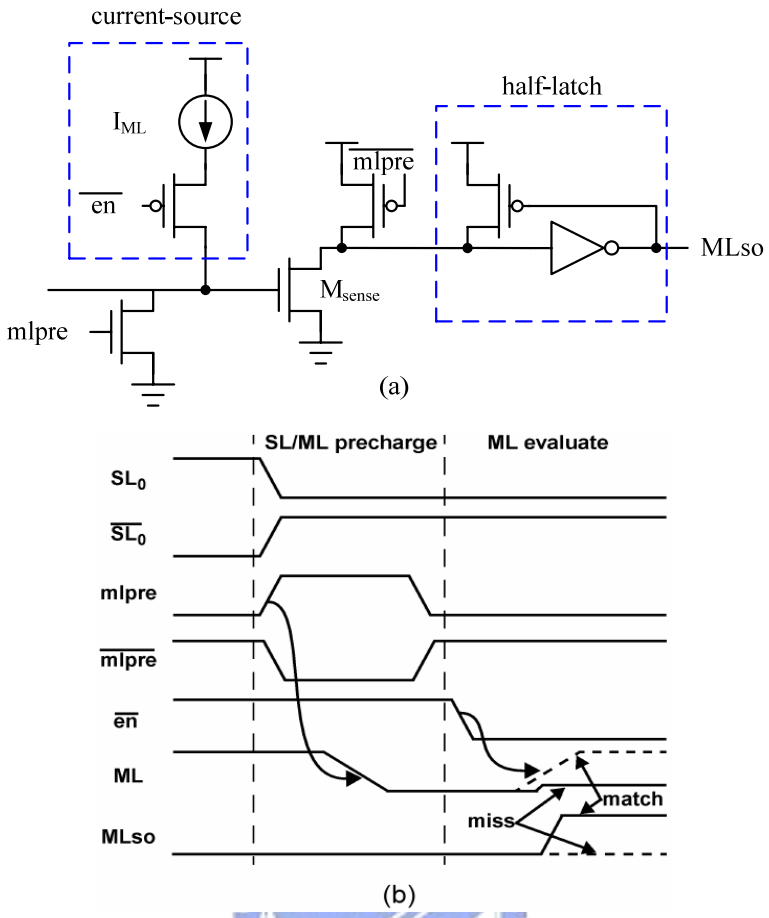

Fig. 2.12 For current-race match-line sensing [48]: (a) a circuit implementation including precharge circuitry and (b) a timing diagram for a single search cycle

2.5.1.3 Selective-Precharge Scheme

Selective-precharge, performs a match operation on the first few bits of a word before activating the search of the remaining bits [49]. For example, in a 144-bit word, selective precharge initially searches only the first 3 bits and then searches the remaining 141 bits only for words that matched in the first 3 bits. Assuming a uniform random data distribution, the initial 3-bit search should allow only 1/2 words to survive to the second stage saving about 88% of the match-line power. In practice, there are two sources of overhead that limit the power saving. First, to maintain speed, the initial match implementation may draw a higher power per bit than the search operation on the remaining bits. Second, an application may have a data distribution that is not uniform, and, in the worst-case scenario, the initial match bits are identical among all words in the CAM, eliminating any power saving. Fig. 2.13 is an example of selective precharge in the original paper [49]. The example uses the first k bits for

the initial search and the remaining m-k bits for the remaining search. The RML is precharged through the PMOS which is controlled by the NAND CAM cell and turned on only if there is a match in the first k CAM bits. The remaining cells are NOR-type cells. Thus, one implementation of selective precharge is to use this mixed NAND/NOR match-line structure. Selective precharge is perhaps the most common method used to save power on match-lines [24], [50-54]. Since it is both simple to implement and can reduce power by a large amount in many CAM applications.

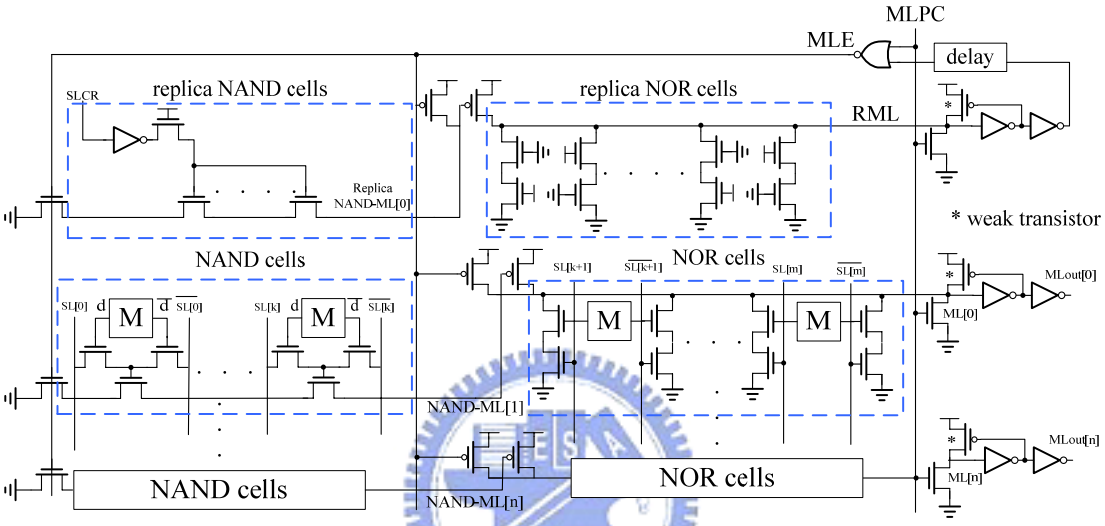

Fig. 2.13 Pulsed NAND-NOR ML architecture

2.5.1.4 Pipelining Scheme

In selective precharge, the match-line is divided into two segments. More generally, the match-line may be divided into any number of segments, where a match in a segment results in a search operation in the next segment but a miss terminates the match operation for that word. A design that uses such style is the pipelined match-lines scheme [55-56]. Fig. 2.14 (a) shows a simplified schematic of a conventional NOR-type match-line structure where all cells are connected in parallel. Fig. 2.14 (b) shows the same set of cells as in Fig. 2.14 (a), but with the match-line broken into four segments that are serially evaluated. If any stage misses, the subsequent stages are shut off, resulting in power saving. The drawbacks of this scheme are the increased latency and the area overhead due to the pipeline stages. A pipelined match-line scheme is not as compelling as basic selective pre-charge; however, pipelining enables the use of hierarchical search-lines, thus saving power.

Another approach is to segment the match-line so that each individual bit forms a segment [57]. Thus, selective pre-charge operates on a bit-by-bit basis. In this design, the CAM cell is modified so that the match evaluation ripples through each CAM cell. If at any cell there is a miss, the subsequent cells do not activate, as there is no need for a comparison operation. The drawback of this scheme is the extra circuitry required at each cell to gate the comparison with the result from the previous cell.

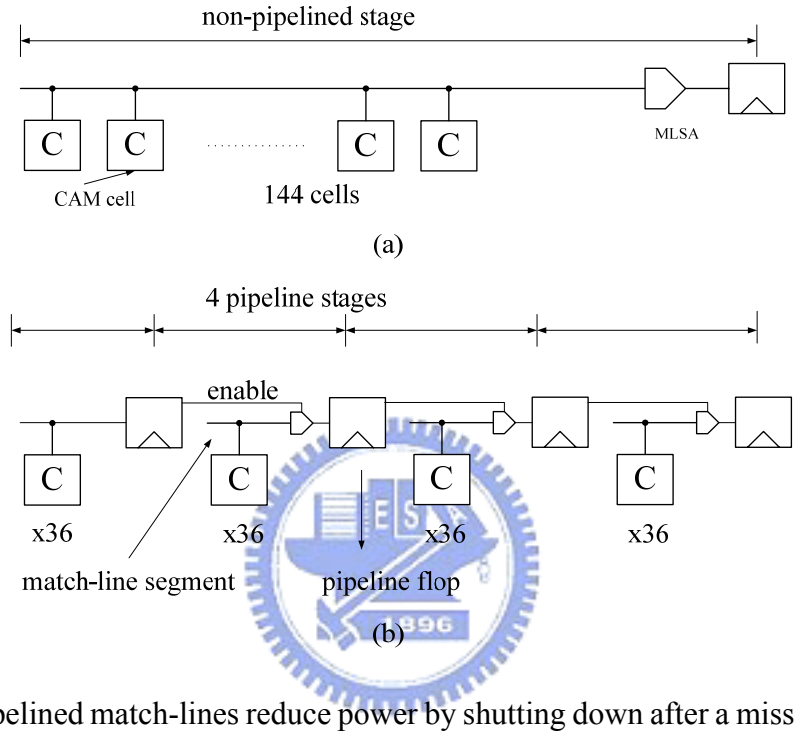

Fig. 2.14 Pipelined match-lines reduce power by shutting down after a miss in a stage

2.5.1.5 Current-Saving Scheme

The current-saving scheme, [58], [59], is another data-dependent match-line-sensing scheme which is a modified form of the current-race sensing scheme. The key improvement of the current-saving scheme is to allocate a different amount of current for a match from a miss. In the current-saving scheme, matches are allocated a larger current and misses are allocated a lower current. Since almost every match-line has a miss, overall the scheme saves power. Fig. 2.15 shows a simplified schematic of the current-saving scheme. The main difference from the current-race scheme is the addition of the current-control block. This block is a mechanism which allocates a different amount of current, based on a match or a miss. The input to this current-control block is the match-line voltage, V_{ML}, and the output is a control voltage that determines the current, I_{ML}, which charges the match-line. The

current-control block provides positive feedback since higher V_{ML} results in higher I_{ML}, which, in turn, results in higher V_{ML}. In this scheme, the match-line is pre-charged low, just as in the current-race scheme. In the current-race scheme, the current source provides a constant amount of current regardless of the state of the match-line. In the current-saving scheme, the amount of current is initially the same for all match-lines, but the current control reduces the current provided to match-lines that miss which have a resistance to ground, but maintains the current to match-lines that match (with no resistance to ground). The current-control block increases the current as the voltage on the ML rises and the voltage on the ML rises faster for large ML resistance. Since the amount of current decreases with the number of misses, the power dissipated on the match-line also depends on the number of bits that miss. References [58] and [59] show that this scheme saves over 50% of match-line power compared to the current-race scheme.

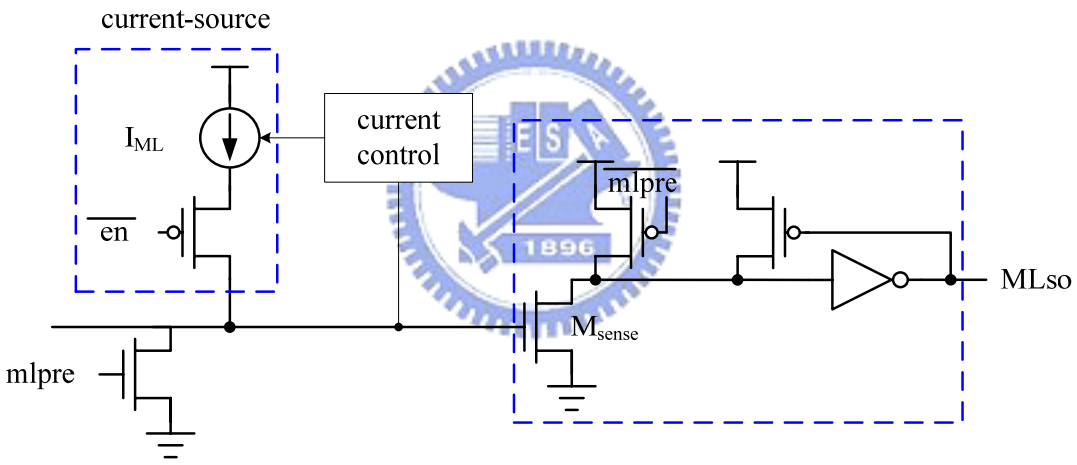

Fig. 2.15 Current-saving matchline-sensing scheme [58], [59]

2.5.2 Method of reducing Search-line power consumption

2.5.2.1 Hierarchical Search-line Scheme

Hierarchical search-lines are built on top of pipelined match-lines [26], [55], [56], [60]. The basic idea of hierarchical search-lines is to exploit the fact that few match-lines survive the first segment of the pipelined match-lines.

With the conventional search-line approach, even though only a small number of match-lines survive the first segment, all search-lines are still driven. Instead of this,

the hierarchical search-line scheme divides the search-lines into a two-level hierarchy of global search-lines (GSLs) and local search-lines (LSLs). Fig. 2.16 shows a simplified hierarchical search-line scheme, where the match-lines are pipelined into two segments, and the search-lines are divided into four LSLs per GSL. In this simplified example, each LSL feeds only a single match-line, but the number of match-lines per LSL can be 64 to 256. The GSLs are active every cycle, but the LSLs are active only when necessary. Activating LSLs is necessary when at least one of the match-lines fed by the LSL is active. In many cases, an LSL will have no active match-lines in a given cycle hence no need to be activated thus saving power. Thus, the overall power consumption on the search-lines is

$$P_{SL} = \left(C_{GSL}V_{DD}^2 + \alpha C_{LSL}V_{DD}^2\right)f \tag{2.6}$$

where C_{GSL} is the GSL capacitance, C_{LSL} is the capacitance of all LSLs connected to a GSL and α is the activity rate of the LSLs. C_{GSL} primarily consists of wiring capacitance, whereas C_{LSL} consists of wiring capacitance and the gate capacitance of the SL inputs of the CAM cells. The factor α, which can be as low as 25% in some cases, is determined by the search data and the data stored in the CAM. (2.6) determines how much power is saved on the LSLs, but the cost is the power caused by the GSLs. Thus, the power dissipated by the GSLs must be sufficiently small so that overall search-line power is lower than that using the conventional approach.

If wiring capacitance is small compared to the parasitic transistor capacitance [60], then the scheme saves power. However, as transistor dimensions scale down, it is expected that wiring capacitance will increase relative to transistor parasitic capacitance. In the situation where wiring capacitance is comparable or larger than the parasitic transistor capacitance, C_{GSL} and C_{LSL} will be similar in size, resulting in no power savings. In this case, small-swing signaling on the GSLs can reduce the power of the GSLs compared to that of the full-swing LSLs. This results in the modified search-line power of

$$P_{SL} = 2n\left(C_{GSL}V_{LOW}^2 + \alpha C_{LSL}V_{DD}^2\right)f \tag{2.7}$$

where V_{LOW} is the low-swing voltage on the GSLs (assuming an externally available power supply V_{LOW}). This scheme requires an amplifier to convert the low-swing GSL signal to the full-swing LSLs signals. Since there is only a small number of these amplifiers per search-line, the area and power overhead can be small.

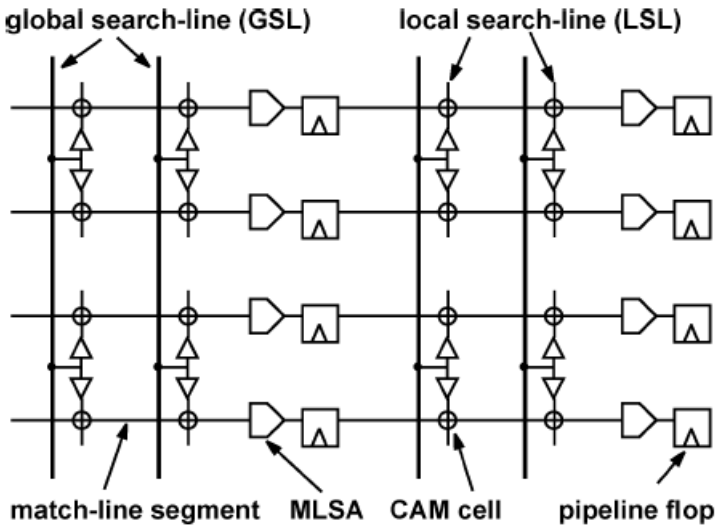

Fig. 2.16 Schematic of the hierarchical search-line architecture.

2.5.2.2 Charge-Recycling Search-line Driver

The SLs consume power only when the search data changes. In general, the transition probability of SL is smaller than one half. The nonprecharged SLs consume power less than half of that of the precharged SLs [55], [56]. The CRSLD further saves the SL power by recycling the charge of SLs. Fig. 2.17 shows the CRSLD architecture. Table 2.6 shows the power comparison of SLs.

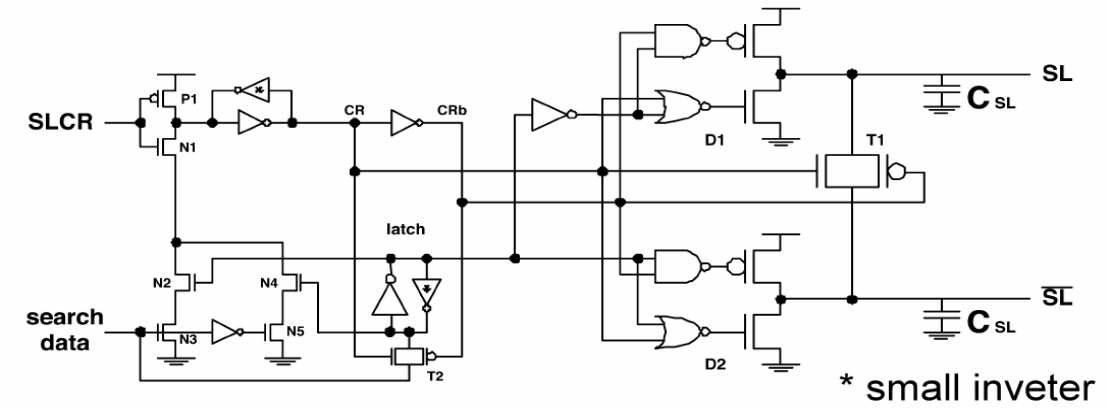

Fig. 2.17 Charge-recycling search-line driver

Table 2.6 Power consumption of CRSLD compare to other SL schemes

	Precharge	Charge recycling	Power of SL
Precharge SL	Yes	No	$f \times C_{SL} \times V_{DD}^2$
Non-precharge SL	No	No	$\alpha \times f \times C_{SL} \times V_{DD}^2$
CRSLD	No	Yes	$\frac{1}{2}\alpha \times f \times C_{SL} \times V_{DD}^2$

2.5.3 Power-Saving CAM architecture

A 32-bit IPv4 destination IP address could be divided into two parts. The first 8-bit is called merged field (MF) and remaining 24-bit is called individual field (IF). Therefore, the hybrid-type TCAM architecture has three different types of banks: the main bank, constructed with high-speed NOR-type TCAM, the sub bank, and the extra bank constructed with low-power AND-type TCAM [62], as shown in Fig. 2.18. In this case, most of the data are divided into common data portion which store in MF and its unique data portion which store IF. The main bank stores k-bit MF, and the number of its word lines is equal to the number of sub bank. The match result of each word line of main bank will activate the corresponding sub bank as illustrated in Fig. 2.19. If some data don't have any common MF, the extra bank stores them without dividing them into MF and IF.

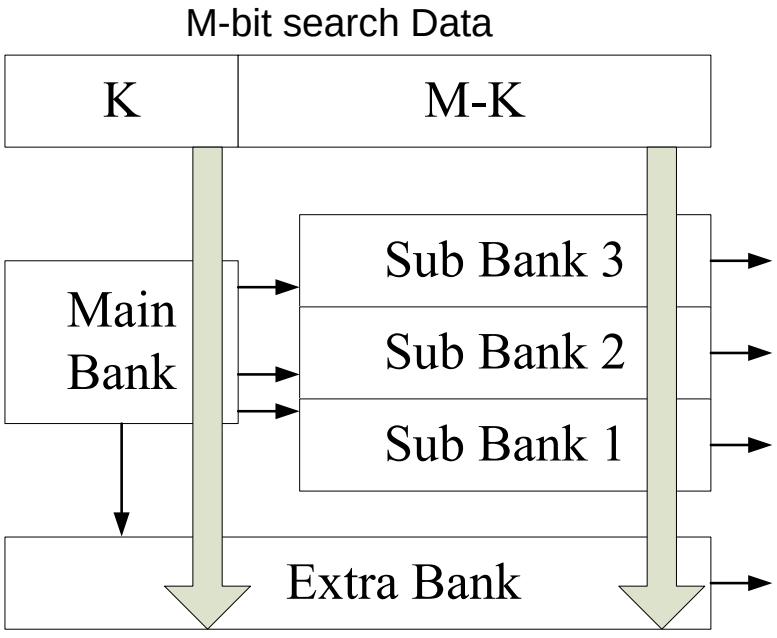

Fig. 2.18 Block diagram of hybrid-type TCAM architecture.

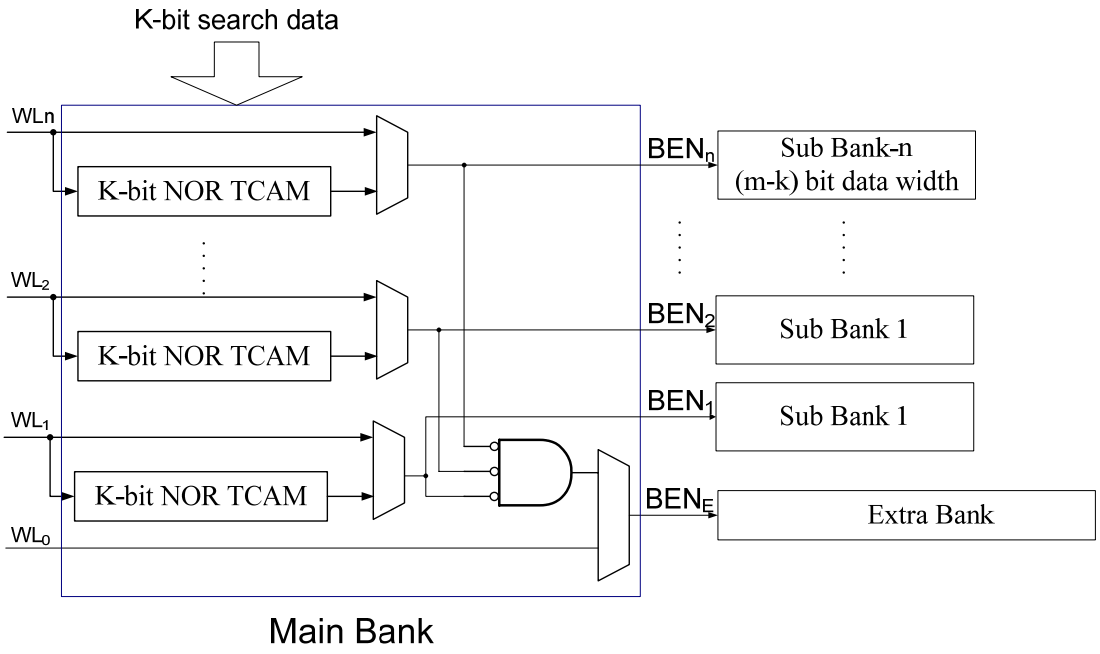

Fig. 2.19 Block diagram of main bank.

Applying this hybrid-type of TCAM architecture, area saving can be expected. Assuming that a conventional CAM of $(B \times m \times W)$ where B is the number of banks, m is the bit width of a bank, and W is the number of words in a bank. However, if applying the hybrid architecture, the total number of the cells is given by:

$$m \times W + (B-1) \times [(m-k) \times W] + (B-1) \times k \qquad (2.8)$$

Therefore, the area is approximately $(m-k)/m$ of the conventional architecture.

In order to reduce the search time and redundant search power consumption, the match-line repeater (MLRPT), 1-bit column decoding, and local priority encoding (LPE) schemes are applied. As described in Fig. 2.20, a word block consists of four cell blocks, two partial match blocks (PM), and a main match block. Each cell block includes 72 AND-type TCAM cells and 3 MLRPT to construct a search engine with 144-bit search data width. 288 cells are attached to each word line and they are column-decodes using LSB address when read or write operation is performed.

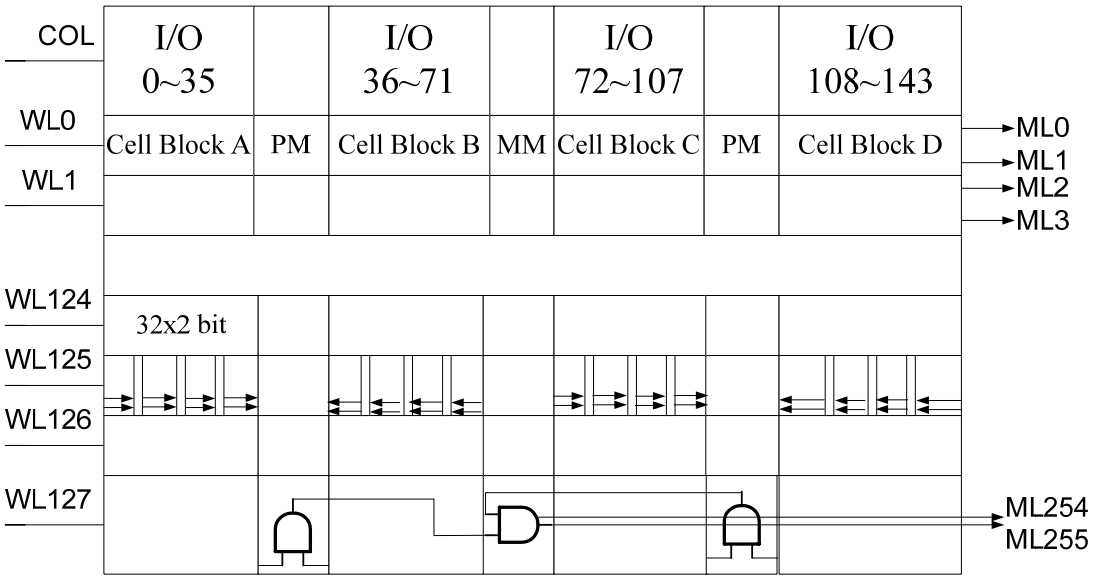

Fig. 2.20 Sub/extra bank structure.

Because the major drawback of AND-type TCAM cell is the slow search time, the match-line repeater is proposed to improve pre-charge and evaluation time. As shown in Fig. 2.21, one MLRPT is composed of pass transistors to speed up the propagation of the match-line.

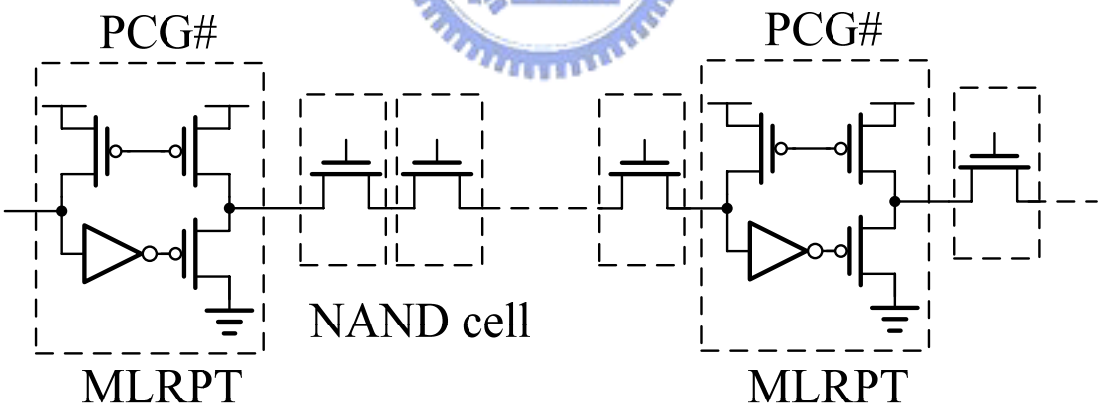

Fig. 2.21 Schematic of match-line repeater.

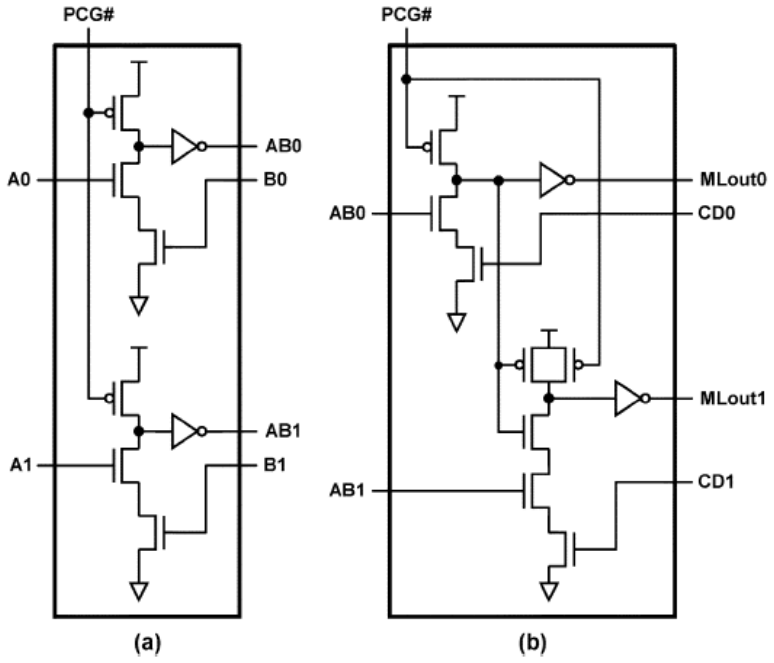

Fig. 2.22 (a) Partial match (PM) block. (b) Main match (MM) block and local priority encoding (LPE).

Fig. 2.22 shows the schematics of partial match block and main match block which also integrates with LPE circuits. AB0, AB1, CD0, and CD1 are evaluated by PMs. (AB0, CD0) and (AB1, CD1) represent the partial match results of column0 and column1, respectively. They are merged to generate the final match results MLout0 and MLout1 by MM. When both column0 and column1 are matched, only the match-line of column0 is valid because of LPE.

2.5.4 Tree Style AND-type Match Lines

AND-type match-line scheme connect CAM cells in serial causes long search time. To decrease the search time, it's necessary to separate the match-line schemes into several segments. Fig. 2.23 shows the four-input AND gate logic transform from dynamic circuits to clock-data pre-charge dynamic (CDPD) circuits [18]. Furthermore, a pseudo-footless clock-and-data pre-charge dynamic (PF-CDPD) circuit [85], which is combined with the AND-type match-line scheme and CDPD circuits are illustrated in Fig. 2.24

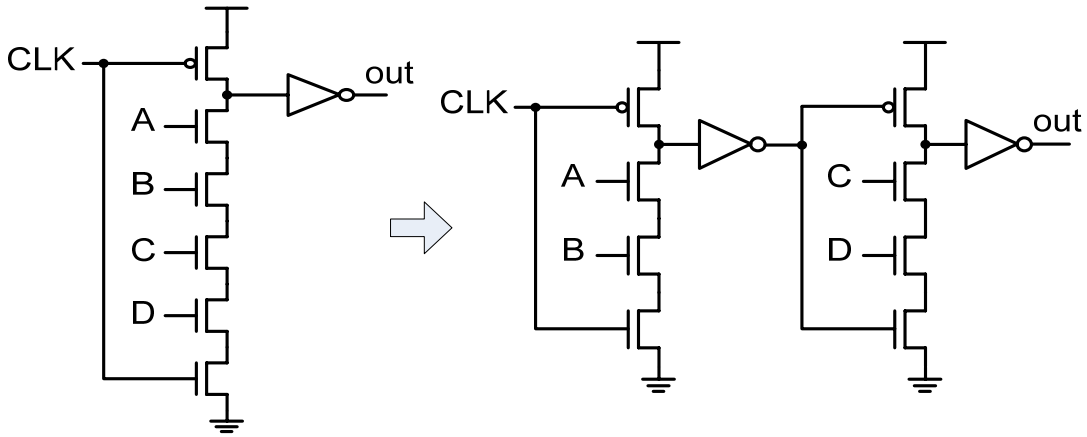

Fig. 2.23 Transfer four-input AND gate logic into clock-and-data pre-charge dynamic (CDPD) circuits.

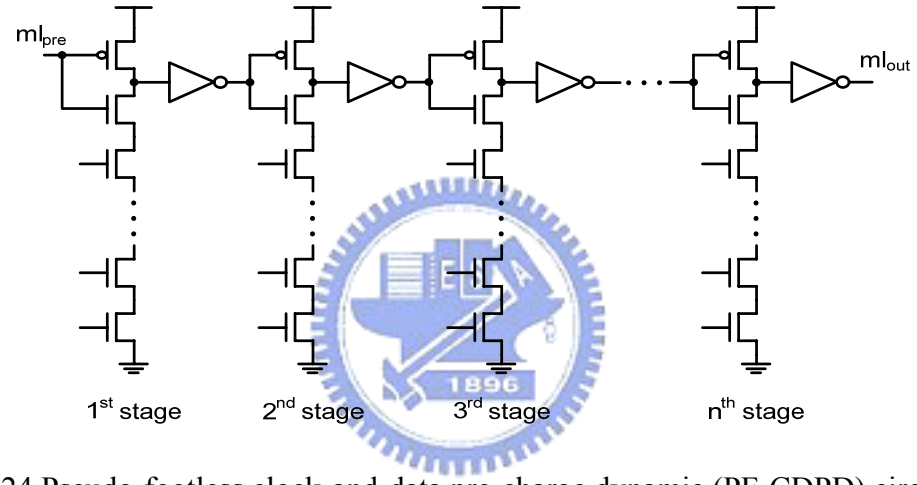

Fig. 2.24 Pseudo-footless clock-and-data pre-charge dynamic (PF-CDPD) circuits.

At the beginning of pre-charge phase, the floating node of first stage is charged to high causing the first stage output becoming low to charge the floating node of next stage. Therefore, all floating nodes are charged to high during pre-charge phase. At the evaluation phase, if the comparisons result of first stage is match, the first output will become low to enable second stage comparison operation. On the other hand, if the comparison result of first stage is mismatch, the output of first stage will still keep low to disable the next stage comparison operation. The slowest evaluation happens when the input data matches with the stored data, and the PF-CDPD behaves much like a series of inverters. Accordingly, there are several advantages of CAM which adopts PF-CDPD circuits. First, the match-lines divided into many segments causes the size of comparison transistor being unnecessary too large in the same search time criteria. Second, the size of comparison transistors becomes smaller, which reduces the switching capacitances, which is also lowered by separated match-line. Third, if

the stored data and search data are mismatch, the output will disable all comparison operations in the rest stages to avoid unnecessary switching. In consequence, PF-CDPD circuits not only enhance search time but also save power consumption.

For further speed-up of the PF-CDPD AND scheme, three versions of tree-style match lines are investigated in [1]. The three versions are shown in. Fig. 2.25. The first one (Fig. 2.25 (b)) has the same depth as the original PF-CDPD design. The AND function is performed by a separated dynamic AND gate except for the first stage. Since these separate gates are not on the critical path, the match speed is improved. However, power consumption is increased due to the requirement of a larger clock driver. The second one adopts a three level tree structure, where the longest path is reduced to 5 PF-CDPD AND gates plus one 6-input static AND gate. Although this scheme does not incur extra power consumption, the speed improvement is quite limited because the 6-input static AND gate is slow and that the interconnections among the dynamic gates induce much RC delay. In lieu of the pros and cons of the previous two approaches, a two-level tree structure is proposed. It can be seen that there is no extra power consumption as the original one and the speed of critical path is higher than the second one due to the larger speed difference between the 6-input and 4-input static AND gates and reduced interconnect RC delay.

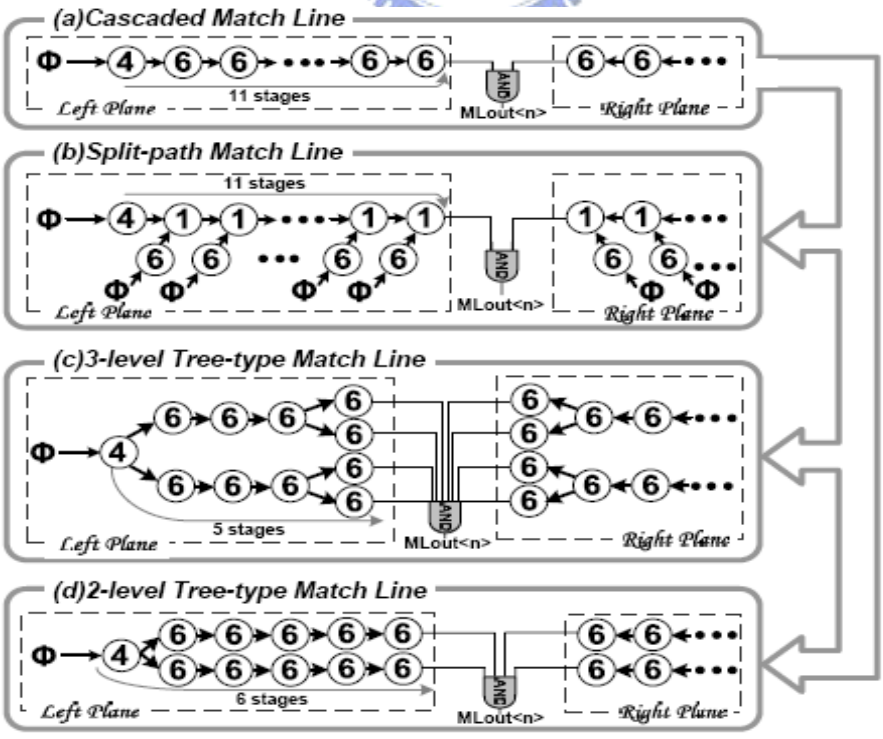

Fig. 2.25 Three tree style match-line scheme

2.6 Method of Reducing Standby Power

2.6.1 Novel Storage Cells for TCAM

[93] proposed a technique to reduce the leakage current of TCAM during standby. In this paper, two novel ternary storage cells that exploit the unique properties of TCAMs for reducing the cell leakage.

Fig. 2.26 shows the leakage paths in the TCAM cell using 6T SRAM cells when the BLs are pre-charged to zero, which minimizes the sub-threshold leakages through the access transistors (N3 to N6) because a TCAM column with shared BLs has the same probability of storing ‘0’, ‘1’ and ‘mask’. The NMOS and PMOS sub-threshold leakages are denoted by ISN and ISP respectively. NMOS gate leakages are specified by IGON and IGOFF for ‘ON’ and ‘OFF’ transistors respectively. Similarly, PMOS gate leakages are expressed by IGONP and IGOFFP. The sub-threshold leakages through the comparison logic transistors are ignored because most modern match line sense amplifiers reset both sides of the comparison logic to ground when they are idle. The gate leakages of the two comparison logic transistors are also ignored because they only depend on the SL data.

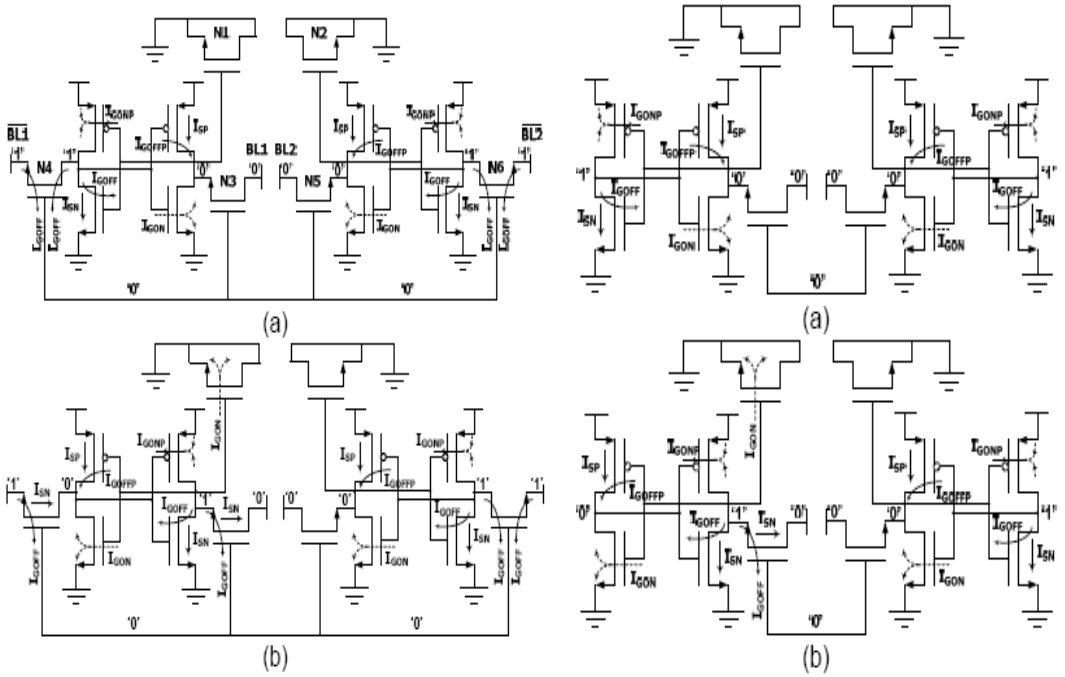

Fig. 2.26 Leakage path of conventional 6T and 5T SRAM cell

Furthermore, a 5T-SRAM cell consumes less leakage and smaller area than a 6T-SRAM cell. The slower READ and WRITE of 5T-SRAM is not an issue for TCAM applications. Thus, 5T-SRAM is an attractive option for leakage and area reduction in TCAMs. Fig. 2.27 shows leakage paths in a 5T-SRAM based TCAM cell. As expected, it has fewer leakage paths than the conventional TCAM cell due to the removal of two access transistors, which also results in smaller cell area.

Since each TCAM cell contains two SRAM cells to store the three states, one state (both N1 and N2 are ‘ON’) is not used. Here a ternary storage cell that trades this unused state for a smaller leakage by coupling two 5T-SRAM cells and eliminating a sub-threshold leakage path is proposed. Fig. 2.27 shows the leakage paths of the proposed NMOS-coupled (NC) ternary storage cell that connects two 5T-SRAM cells using NMOS transistors. Under ‘0’ and ‘1’ conditions (Fig. 2.27(b)), one of the coupling NMOS transistors keeps the storage node at ‘0’ by coupling it to the ‘0’ of the other 5T-SRAM cell. The other NMOS transistor (N7) is ‘OFF’ but it does not contribute sub-threshold leakage because its source and drain both are at logic ‘1’. Thus, this cell will exhibit smaller leakage than the 5T-SRAM based cell if the sub-threshold leakage is much larger than the gate leakage.

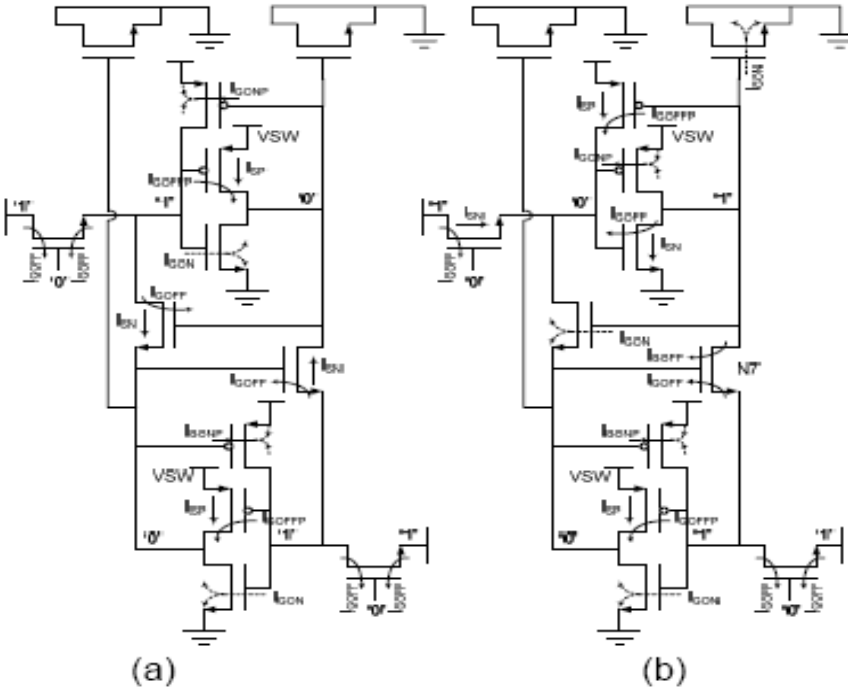

Fig. 2.27 N-type Combined CAM Architecture

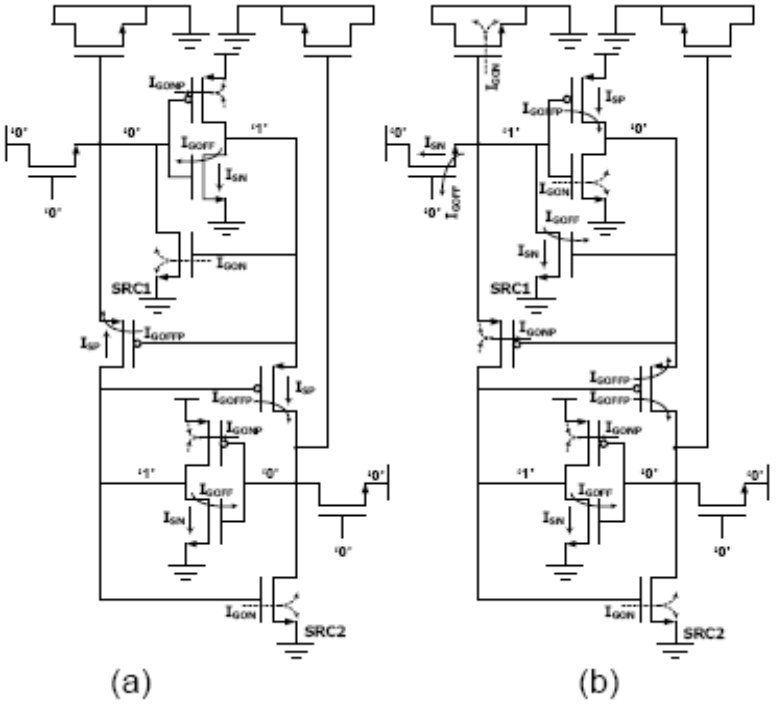

Fig. 2.28 P-type Combined CAM Architecture

The coupling between the two 5T-SRAM cells can also be achieved by PMOS transistors, and the proposed PMOS coupled (PC) cell and its leakage path is shown in Fig. 2.28. Under '0' and '1' conditions, one of the coupling PMOS transistors does not consume sub-threshold leakage similar to the NCTCAM cell. Thus, it will also exhibit smaller leakage than the 5T-SRAM based cell if the sub-threshold leakage is much larger than the gate leakage.