Testing and Diagnosis Methodologies for Embedded CAM

Test Algorithm

Test Element

- Consists of a number of CAM Operations
- With a prespecified address sequence

ascending (⇑), descending (⇓)

either way(⇕)

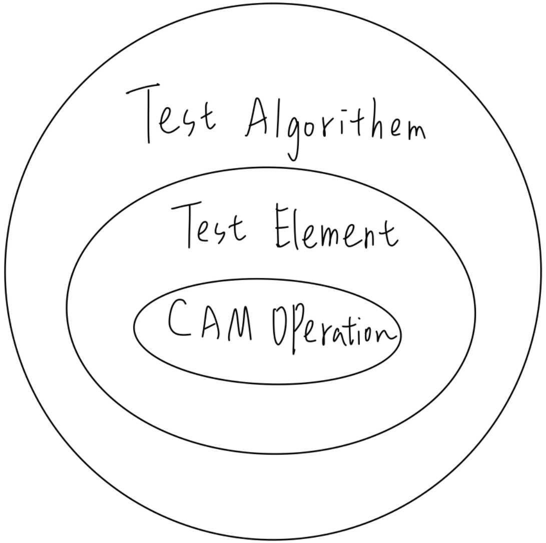
CAM Operations

wD • Write an input Pattern D

rD • read an expected data D

cP_D^M • given the mask pattern M, compare an input pattern D with all words in the CAM

E (Erase) • invalidate the word



March – Like Test

$$\text{MLT-1} = \{\Updownarrow(w\mathbf{1}); \Uparrow(w\mathbf{0}, cP_{\mathbf{0}}; w\mathbf{1}); \Uparrow(r\mathbf{1}, w\mathbf{0}); \left(cP_{\mathbf{1}}^{\omega(0)}, cP_{\mathbf{1}}^{\omega(1)}, \ldots, cP_{\mathbf{1}}^{\omega(W-1)}\right); \Downarrow(w\mathbf{1}, cP_{\mathbf{1}}, w\mathbf{0}); \Downarrow(r\mathbf{0}, w\mathbf{1}); \left(cP_{\mathbf{0}}^{\omega(0)}, cP_{\mathbf{0}}^{\omega(1)}, \ldots, cP_{\mathbf{0}}^{\omega(W-1)}\right)\}$$

$$\text{MLT-2} = \{\Updownarrow(wD); \Updownarrow(w\bar{D}); \left(cP_{D}^{\omega(0)}, cP_{D}^{\omega(1)}, \ldots, cP_{D}^{\omega(W-1)}\right); \Updownarrow(wD); \left(cP_{\bar{D}}^{\omega(0)}, cP_{\bar{D}}^{\omega(1)}, \ldots, cP_{\bar{D}}^{\omega(W-1)}\right)\}$$

$$\omega(i) = 2^W - 1 - 2^i$$

$$\text{MLT-1} = \{\Updownarrow(w\mathbf{1}); \Uparrow(w\mathbf{0}, cP_{\mathbf{0}}; w\mathbf{1}); \Uparrow(r\mathbf{1}, w\mathbf{0}); \\ (cP_{\mathbf{1}}^{\omega(0)}, cP_{\mathbf{1}}^{\omega(1)}, \ldots, cP_{\mathbf{1}}^{\omega(W-1)}); \Downarrow(w\mathbf{1}, cP_{\mathbf{1}}, w\mathbf{0}); \\ \Downarrow(r\mathbf{0}, w\mathbf{1}); (cP_{\mathbf{0}}^{\omega(0)}, cP_{\mathbf{0}}^{\omega(1)}, \ldots, cP_{\mathbf{0}}^{\omega(W-1)})\}$$

1. Initializes the CAM array to the all – 1 solid background

		Content (M_i)		
Operation		W_0 addressed	W_1 addressed	W_2 addressed
$w\mathbf{1}$	W_0	111 (x)	111 (x)	111 (x)
	W_1	111 (x)	111 (x)	111 (x)
	W_2	111 (x)	111 (x)	111 (x)

$$\text{MLT-1} = \{\Updownarrow(w\mathbf{1}); \Uparrow(w\mathbf{0}, cP_{\mathbf{0}}; w\mathbf{1}); \Uparrow(r\mathbf{1}, w\mathbf{0}); (cP_{\mathbf{1}}^{\omega(0)}, cP_{\mathbf{1}}^{\omega(1)}, \ldots, cP_{\mathbf{1}}^{\omega(W-1)}); \Downarrow(w\mathbf{1}, cP_{\mathbf{1}}, w\mathbf{0}); \Downarrow(r\mathbf{0}, w\mathbf{1}); (cP_{\mathbf{0}}^{\omega(0)}, cP_{\mathbf{0}}^{\omega(1)}, \ldots, cP_{\mathbf{0}}^{\omega(W-1)})\}$$

2. $\Uparrow(w\mathbf{0}, cP_{\mathbf{0}}; w\mathbf{1})$

Table 2. Fault-free status when element $\Uparrow (w\mathbf{0}, cP_{\mathbf{0}}, w\mathbf{1})$ is executed.

Operation		Content (M_i)		
		W_0 addressed	W_1 addressed	W_2 addressed
$w\mathbf{0}$	W_0	000 (x)	111 (x)	111 (x)
	W_1	111 (x)	000 (x)	111 (x)
	W_2	111 (x)	111 (x)	000 (x)
$cP_{\mathbf{0}}$	W_0	000 (1)	111 (0)	111 (0)
	W_1	111 (0)	000 (1)	111 (0)
	W_2	111 (0)	111 (0)	000 (1)
$w\mathbf{1}$	W_0	111 (x)	111 (x)	111 (x)
	W_1	111 (x)	111 (x)	111 (x)
	W_2	111 (x)	111 (x)	111 (x)

- don’t-care (x) state during the Write or Read operation.
- The cP0 operation compares the all-0 pattern with the contents of all words

(i.e., W0,W1, and W2) in the CAM concurrently

- fault free: M0 = 1, and Hit = 1.
- SAF, interword CFs,
- SMMF, PMF, EMMF, XMMF, and SIVF

$$\text{MLT-1} = \{\Updownarrow(w\mathbf{1}); \Uparrow(w\mathbf{0}, cP_{\mathbf{0}}; w\mathbf{1}); \Uparrow(r\mathbf{1}, w\mathbf{0}); (cP_{\mathbf{1}}^{\omega(0)}, cP_{\mathbf{1}}^{\omega(1)}, \ldots, cP_{\mathbf{1}}^{\omega(W-1)}); \Downarrow(w\mathbf{1}, cP_{\mathbf{1}}, w\mathbf{0}); \Downarrow(r\mathbf{0}, w\mathbf{1}); (cP_{\mathbf{0}}^{\omega(0)}, cP_{\mathbf{0}}^{\omega(1)}, \ldots, cP_{\mathbf{0}}^{\omega(W-1)})\}$$

3. $\Uparrow(r\mathbf{1}, w\mathbf{0})$

Table 3. Fault-free status when element $\Uparrow(r\mathbf{1}, w\mathbf{0})$ is executed.

Operation		Content (M_i)		
		W_0 addressed	W_1 addressed	W_2 addressed
$r\mathbf{1}$	W_0	111 (x)	000 (x)	000 (x)
	W_1	111 (x)	111 (x)	000 (x)
	W_2	111 (x)	111 (x)	111 (x)
$w\mathbf{0}$	W_0	000 (x)	000 (x)	000 (x)
	W_1	111 (x)	000 (x)	000 (x)
	W_2	111 (x)	111 (x)	000 (x)

- w0 and w1: ↓ and ↑ transitions
- r1: inter-word CFid (Aggressors Higher)
- w0: ↓ transitions
- r1: inter-word CFid (Aggressors Lower)
- SAF(0)
- inter-word CFst where the aggressors are in state 0 and at lower addresses than the victims or in state 1 and at higher addresses

$$\text{MLT-1} = \{\Updownarrow(w\mathbf{1}); \Uparrow(w\mathbf{0}, cP_{\mathbf{0}}; w\mathbf{1}); \Uparrow(r\mathbf{1}, w\mathbf{0});$$
$$\left(cP_{\mathbf{1}}^{\omega(0)}, cP_{\mathbf{1}}^{\omega(1)}, \ldots, cP_{\mathbf{1}}^{\omega(W-1)}\right); \Downarrow(w\mathbf{1}, cP_{\mathbf{1}}, w\mathbf{0});$$
$$\Downarrow(r\mathbf{0}, w\mathbf{1}); \left(cP_{\mathbf{0}}^{\omega(0)}, cP_{\mathbf{0}}^{\omega(1)}, \ldots, cP_{\mathbf{0}}^{\omega(W-1)}\right)\}$$

4. $\left(cP_{\mathbf{1}}^{\omega(0)}, cP_{\mathbf{1}}^{\omega(1)}, \ldots, cP_{\mathbf{1}}^{\omega(W-1)}\right)$

Table 4. Fault-free status when element $(cP_{\mathbf{1}}^{\omega(0)}, cP_{\mathbf{1}}^{\omega(1)}, \ldots, cP_{\mathbf{1}}^{\omega(W-1)})$ is executed.

	Content (M_i)		
	$cP_{\mathbf{1}}^{6}$	$cP_{\mathbf{1}}^{5}$	$cP_{\mathbf{1}}^{3}$
W_0	000 (0)	000 (0)	000 (0)
W_1	000 (0)	000 (0)	000 (0)
W_2	000 (0)	000 (0)	000 (0)

- fault-free: Hit = 0, no 1 in that column
- SAF(0), SAF(1) and the CF aggressor is 0 or undergoes a ↓ transition to turn the victim into 1.
- SMF, CMF, and IMF

MLT-1 Shortcoming

intra-word CFst and CFid are not fully covered by MLT-1, since only the solid (all-0 and all-1) backgrounds are used

$$\text{MLT-2} = \Big\{ \Updownarrow (w D); \Updownarrow (w \bar{D}); \\ \left(c P_{D}^{\omega(0)}, c P_{D}^{\omega(1)}, \ldots, c P_{D}^{\omega(W-1)}\right); \\ \Updownarrow (w D); \left(c P_{\bar{D}}^{\omega(0)}, c P_{\bar{D}}^{\omega(1)}, \ldots, c P_{\bar{D}}^{\omega(W-1)}\right) \Big\}$$

- supplied by different data backgrounds
- the faulty cell is changed after one of the Write operations
- 3 and 5 test element (observe)
- fault free: Hit = 0

$3N \log_2 W$ Write operations

$2W \log_2 W$ Compare operations

Diagonostic Algorithms

- The faulty cells in a word (row)

$$\text{FLR-0} = \{\Updownarrow(E);\, w\mathbf{0};\, (cP_{\mathbf{0}}^{\omega(0)}, cP_{\mathbf{0}}^{\omega(1)}, \ldots, cP_{\mathbf{0}}^{\omega(W-1)})\}$$

$$\text{FLR-1} = \{\Updownarrow(E);\, w\mathbf{1};\, (cP_{\mathbf{1}}^{\omega(0)}, cP_{\mathbf{1}}^{\omega(1)}, \ldots, cP_{\mathbf{1}}^{\omega(W-1)})\}$$

- faulty cells in a column

$$\text{FLC-0} = \{\Updownarrow(E);\, \Updownarrow(w\mathbf{0}, cP_{\mathbf{1}}^{M}, E)\}$$

$$\text{FLC-1} = \{\Updownarrow(E);\, \Updownarrow(w\mathbf{1}, cP_{\mathbf{0}}^{M}, E)\}$$

$$\omega(i) = 2^{W} - 1 - 2^{i}$$

$$\Uparrow (w\mathbf{0}, cP_{\mathbf{0}}; w\mathbf{1})$$

$$\text{FLR-0} = \{\Updownarrow (E); w\mathbf{0}; (cP_{\mathbf{0}}^{\omega(0)}, cP_{\mathbf{0}}^{\omega(1)}, \ldots, cP_{\mathbf{0}}^{\omega(W-1)})\}$$

$$\text{FLR-1} = \{\Updownarrow (E); w\mathbf{1}; (cP_{\mathbf{1}}^{\omega(0)}, cP_{\mathbf{1}}^{\omega(1)}, \ldots, cP_{\mathbf{1}}^{\omega(W-1)})\}$$

$$\Downarrow (w\mathbf{1}, cP_{\mathbf{1}}, w\mathbf{0});$$

$$\text{MLT-1} = \{\Updownarrow (w\mathbf{1}); \Uparrow (w\mathbf{0}, cP_{\mathbf{0}}; w\mathbf{1}); \Uparrow (r\mathbf{1}, w\mathbf{0}); (cP_{\mathbf{1}}^{\omega(0)}, cP_{\mathbf{1}}^{\omega(1)}, \ldots, cP_{\mathbf{1}}^{\omega(W-1)}); \Downarrow (w\mathbf{1}, cP_{\mathbf{1}}, w\mathbf{0}); \Downarrow (r\mathbf{0}, w\mathbf{1}); (cP_{\mathbf{0}}^{\omega(0)}, cP_{\mathbf{0}}^{\omega(1)}, \ldots, cP_{\mathbf{0}}^{\omega(W-1)})\}$$

- E: first resets all valid bits
- w0: sets its valid bit
- Note that the w0 operation does not change the state of the faulty cells (victims) since other words are not changed
- Fault free: Hit = 1

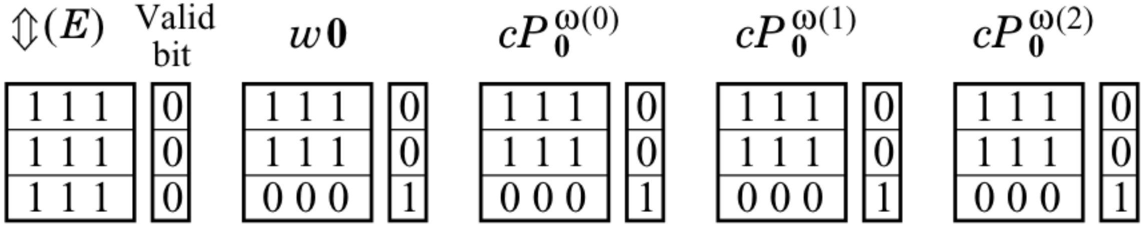


Fig. 3. An example for locating the faulty cells in W_2.

$$\left(cP_1^{\omega(0)}, cP_1^{\omega(1)}, \ldots, cP_1^{\omega(W-1)}\right)$$

$$\text{FLC-0} = \left\{ \Updownarrow (E); \Updownarrow \left(w\mathbf{0}, cP_{\mathbf{1}}^{M}, E\right)\right\}$$

$$\text{FLC-1} = \left\{ \Updownarrow (E); \Updownarrow \left(w\mathbf{1}, cP_{\mathbf{0}}^{M}, E\right)\right\}$$

$$\left(cP_{\mathbf{0}}^{\omega(0)}, cP_{\mathbf{0}}^{\omega(1)}, \ldots, cP_{\mathbf{0}}^{\omega(W-1)}\right)$$

$$\text{MLT-1} = \{\Updownarrow (w\mathbf{1}); \Uparrow (w\mathbf{0}, cP_{\mathbf{0}}; w\mathbf{1}); \Uparrow (r\mathbf{1}, w\mathbf{0});$$
$$\left(cP_{\mathbf{1}}^{\omega(0)}, cP_{\mathbf{1}}^{\omega(1)}, \ldots, cP_{\mathbf{1}}^{\omega(W-1)}\right); \Downarrow (w\mathbf{1}, cP_{\mathbf{1}}, w\mathbf{0});$$
$$\Downarrow (r\mathbf{0}, w\mathbf{1}); \left(cP_{\mathbf{0}}^{\omega(0)}, cP_{\mathbf{0}}^{\omega(1)}, \ldots, cP_{\mathbf{0}}^{\omega(W-1)}\right)\}$$

- E: first resets all valid bits
- w0: write an all-0 patterns words by words
- Note that the w0 operation does not change the state of the faulty cells (victims) since other words are not changed
- Fault free: Hit = 0

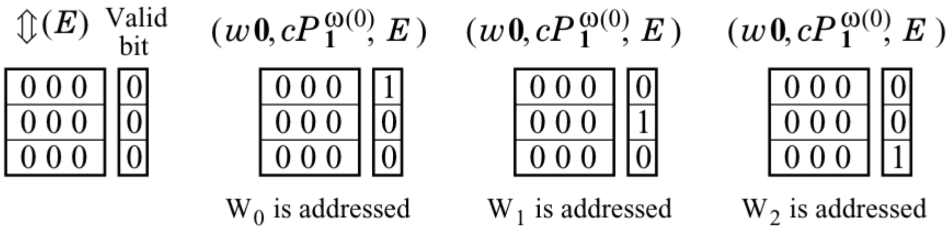


Fig. 4. An example for locating the faulty cells in column 0.

Table 5. Fault coverage comparison for three test algorithms.

	SMF	SMMF	CMF	PMF	EMMF	IMF	XMMF	XMF	SIVF	SVF	MSAF
T_{CAM}	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
NCDA	100%	100%	100%	100%	100%	100%	100%	0%	100%	0%	0%
MLT-1	100%	100%	100%	100%	100%	100%	100%	0%	100%	0%	0%
$T_{CAM}{}^*$	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
NCDA*	100%	$1/N\%$	100%	100%	$1/N\%$	100%	$1/N\%$	0%	100%	0%	0%
MLT-1*	100%	100%	100%	100%	100%	100%	100%	0%	100%	0%	0%

Table 6. Fault coverage statistics for RAM faults.

	SAF	TF	CFst	CFid
T_{CAM}	100%	100%	90%	20%
NCDA	100%	100%	90%	90%
MLT-1	100%	100%	90%	90%
MLT-1 + MLT-2	100%	100%	100%	100%

Note that in this paper we do not provide algorithms for detecting XMF, XVF, and MSAF, since they are covered by the existing efficient test algorithms T_{XM}, T_{SV} , and T_{SU}, respectively [5].

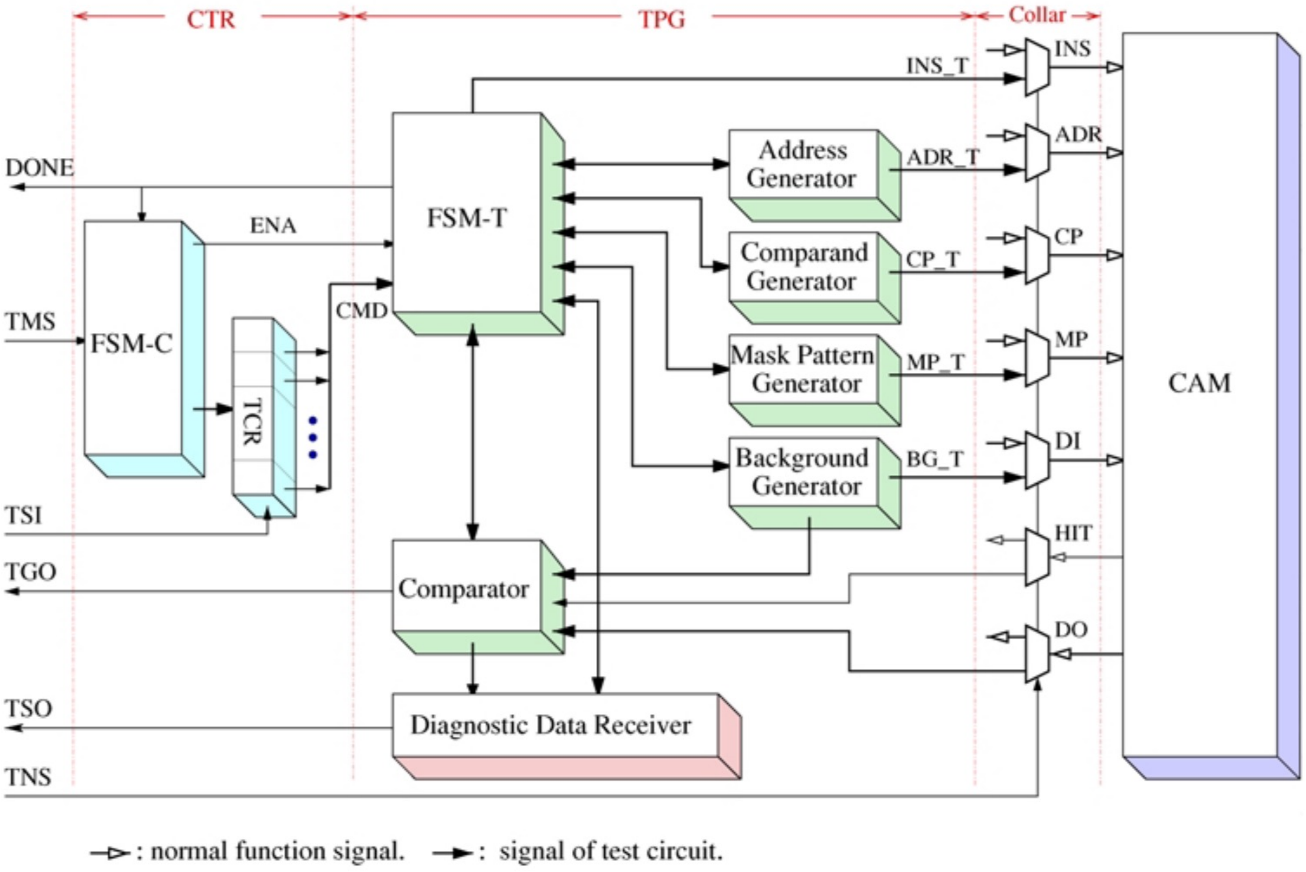


Fig. 5. Block diagram of the proposed BIST design.

R
W
C

Conclusion

- high fault coverage and low time complexity
- observed only by the Hit output or the priority encoder output.
- diagnostic algorithms: positions of the faulty cells
- Future: diagnostic algorithms that can also distinguish between different fault types.