

# Mapping: Fully-Associative Mapping

- Main Memory Block이 Cache의 어떤 Slot으로든 적재 가능
- Tag Field = Main Memory Block 번호

Main Memory Address

Tag(5)	Word(2)	Data(32)
00000	00	'comp'
00001	00	'memo'
00010	00	'cute'
00011	00	'very'
00100	00	'much'
00101	00	'kent'
00110	00	'copy'
00111	00	'poor'
01000	00	'sang'
01001	00	'role'
01010	00	'cach'
01011	00	'full'
01100	00	'sign'
01101	00	'univ'
01110	00	'star'
01111	00	'digi'
10000	00	'star'
10001	00	'para'
10010	00	'fpga'
10011	00	'arch'
10100	00	'vhdl'
10101	00	'edge'
10110	00	'mimo'
10111	00	'syst'
11000	00	'time'
11001	00	'evol'
11010	00	'mooc'
11011	00	'soci'
11100	00	'code'
11101	00	'comm'
11110	00	'wire'
11111	00	'enda'

32비트  
주기억장치  
(총 128바이트)

Tag	Data	Slot 번호
00001	'memo'	0(000)
00110	'copy'	1(001)
		2(010)
		3(011)
		4(100)
		5(101)
		6(110)
		7(111)

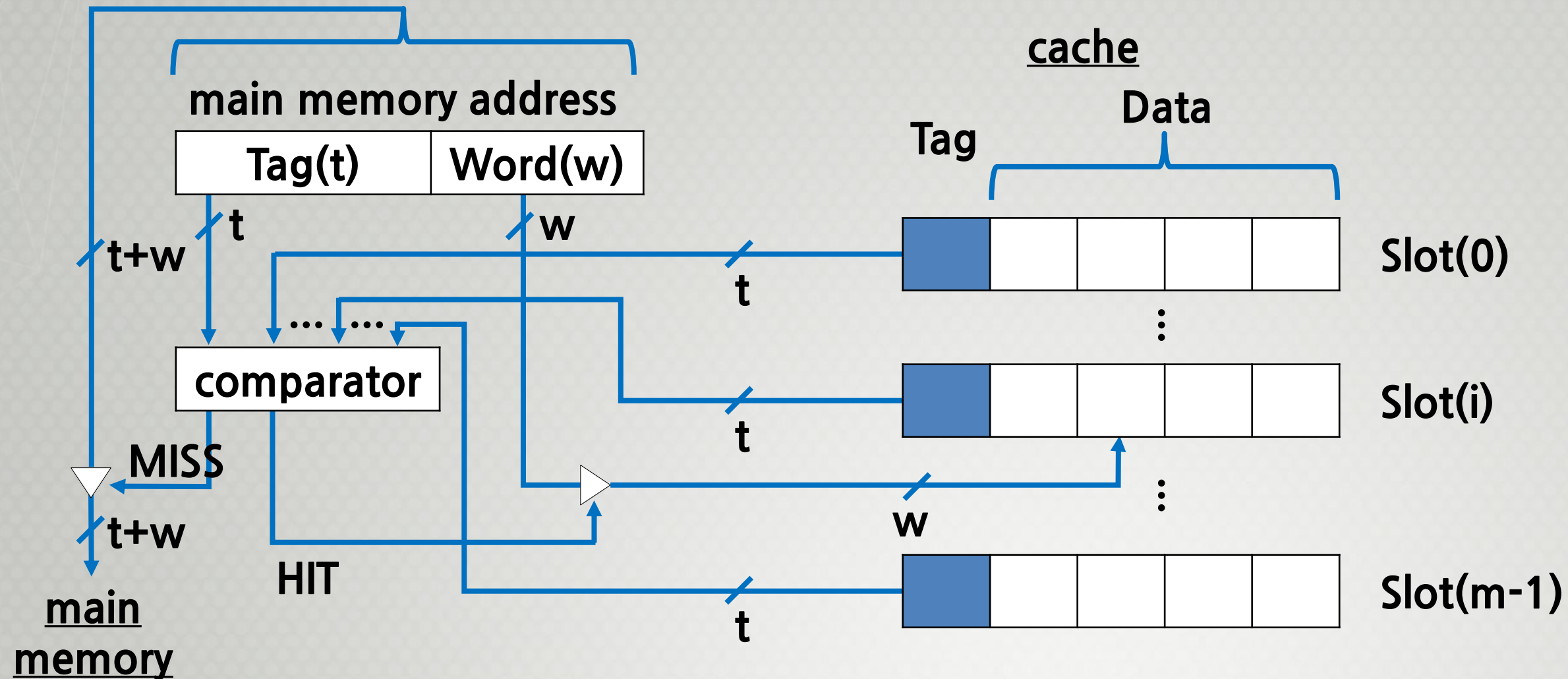
5비트      32비트  
캐시(32바이트)

00001 00 → Hit  
 00011 01 → Miss후 적재  
 00110 10 → Hit  
 01111 00 → Miss후 적재  
 11110 11 → Miss후 적재

Tag	Data	Slot 번호
00001	'memo'	0(000)
00110	'copy'	1(001)
00011	'very'	2(010)
01111	'digi'	3(011)
11110	'wire'	4(100)
		5(101)
		6(110)
		7(111)

5비트      32비트  
캐시(32바이트)

# Mapping: Fully-Associative Mapping-H/W 조직



## 장점

- 새로운 Block이 Cache로 적재될 때 **Slot의 선택이 매우 자유로움**
- Locality이 높다면, **Hit Ratio가 매우 높아짐**

## 단점

- Cache Slot들의 Tag들을 병렬로 검사하기 위하여 **매우 복잡하고 비용이 높은 회로 필요**



# Mapping: Direct Mapping

Main Memory Address

Tag(2)	Slot(3)	Word(2)	Data(32)
00	000	00	'comp'
00	001	00	'memo'
00	010	00	'cute'
00	011	00	'very'
00	100	00	'much'
00	101	00	'kent'
00	110	00	'copy'
00	111	00	'poor'
01	000	00	'sang'
01	001	00	'role'
01	010	00	'cach'
01	011	00	'full'
01	100	00	'sign'
01	101	00	'univ'
01	110	00	'star'
01	111	00	'digi'
10	000	00	'star'
10	001	00	'para'
10	010	00	'fpga'
10	011	00	'arch'
10	100	00	'vhdl'
10	101	00	'edge'
10	110	00	'mimo'
10	111	00	'syst'
11	000	00	'time'
11	001	00	'evol'
11	010	00	'mooc'
11	011	00	'soci'
11	100	00	'code'
11	101	00	'comm'
11	110	00	'wire'
11	111	00	'enda'

32비트  
주기억장치  
(총 128바이트)

- Main Memory Block이 Cache의 정해진 Slot으로만 적재 가능
- Tag Field + Slot Field = Main Memory Block 번호

Cache Slot	Main memory block 번호			
0(000)	00 000	01 000	10 000	11 000
1(001)	00 001	01 001	10 001	11 001
2(010)	00 010	01 010	10 010	11 010
3(011)	00 011	01 011	10 011	11 011
4(100)	00 100	01 100	10 100	11 100
5(101)	00 101	01 101	10 101	11 101
6(110)	00 110	01 110	10 110	11 110
7(111)	00 111	01 111	10 111	11 111

Tag	Data	Slot 번호
00	'comp'	0(000)
10	'para'	1(001)
		2(010)
		3(011)
		4(100)
		5(101)
		6(110)
		7(111)

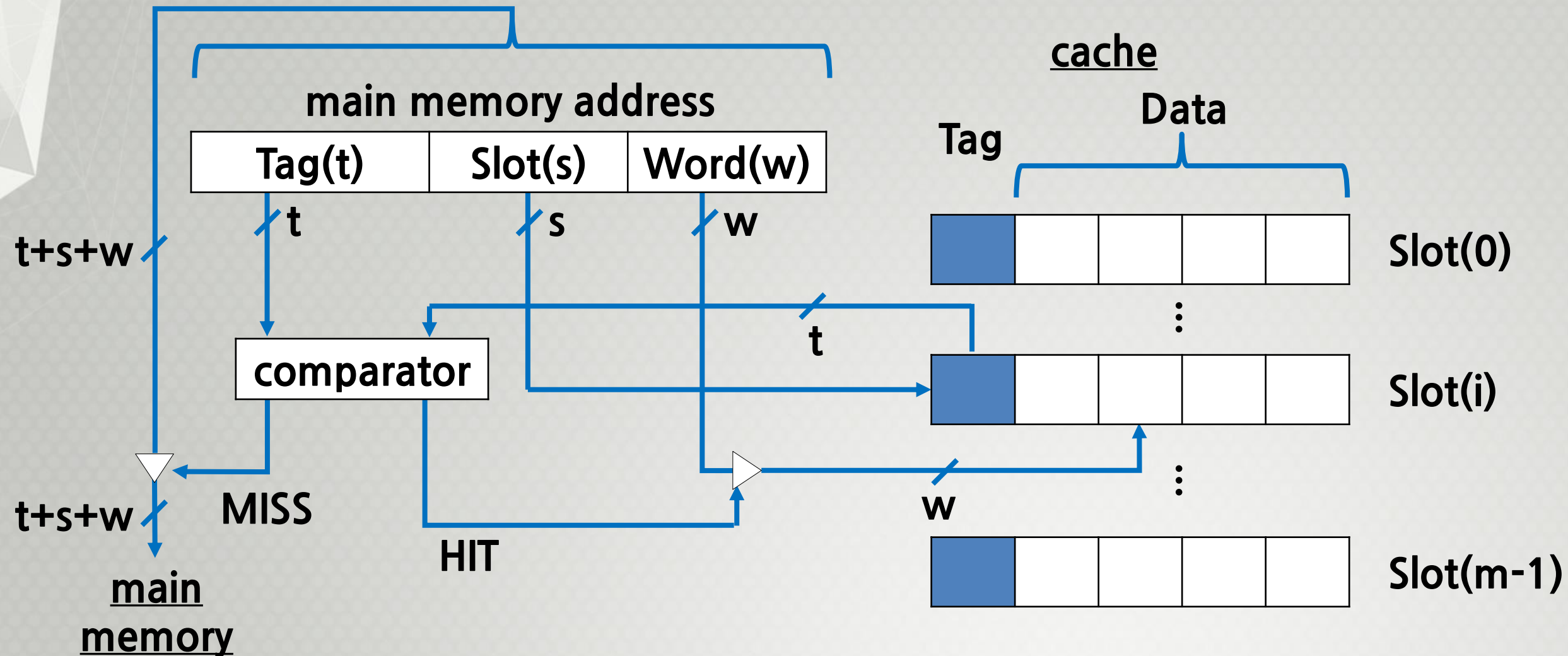
2비트      32비트  
캐시(32바이트)

00 001 00 → Miss후 적재  
00 000 00 → Hit  
00 110 00 → Miss후 적재  
01 111 00 → Miss후 적재

Tag	Data	Slot 번호
00	'comp'	0(000)
00	'memo'	1(001)
		2(010)
		3(011)
		4(100)
		5(101)
		6(110)
		7(111)

2비트      32비트  
캐시(32바이트)

# Mapping: Direct Mapping-H/W 조직



## 장점

- H/W가 간단하고, 구현 비용이 적게 듭

## 단점

- 각 MM Block이 적재될 수 있는 Cache Slot이 한 개뿐이기 때문에 그 Slot을 Share하는 다른 블록이 적재되는 경우에는 **Swap-out** 됨



# Mapping: Set-Associative Mapping

Main Memory Address			Data(32)
Tag(3)	Set(2)	Word(2)	
000	00	00	'comp'
000	01	00	'memo'
000	10	00	'cute'
000	11	00	'very'
001	00	00	'much'
001	01	00	'kent'
001	10	00	'copy'
001	11	00	'poor'
010	00	00	'sang'
010	01	00	'role'
010	10	00	'cach'
010	11	00	'full'
011	00	00	'sign'
011	01	00	'univ'
011	10	00	'star'
011	11	00	'digi'
100	00	00	'star'
100	01	00	'para'
100	10	00	'fpga'
100	11	00	'arch'
101	00	00	'vhdl'
101	01	00	'edge'
101	10	00	'mimo'
101	11	00	'syst'
110	00	00	'time'
110	01	00	'evol'
110	10	00	'mooc'
110	11	00	'soci'
111	00	00	'code'
111	01	00	'comm'
111	10	00	'wire'
111	11	00	'enda'

32비트  
주기억장치  
(총 128바이트)

- Set Associative Mapping Cache = Direct Mapping + Fully-Associative Mapping
- Main Memory Block이 Cache의 정해진 Slot으로만 적재 가능(Tag Field + Slot Field = Main Memory Block 번호)

Tag	Data	Tag	Data	Set 번호
				0(00)
000	'memo'			1(01)
001	'copy'			2(10)
				3(11)

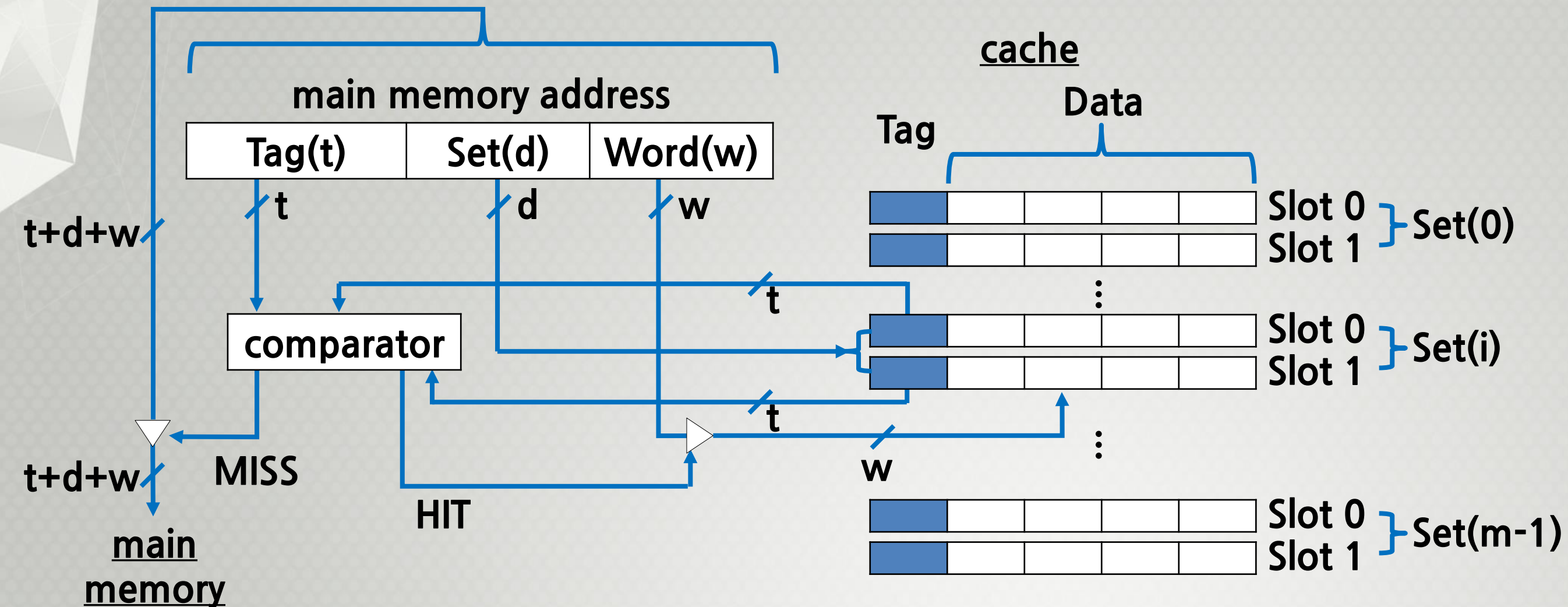
3비트 32비트 3비트 32비트  
캐시(32바이트)

000 01 00 → Hit  
 000 11 00 → Miss후 적재  
 001 10 00 → Hit  
 011 11 00 → Miss후 적재  
 111 10 00 → Miss후 적재

Tag	Data	Tag	Data	Set 번호
				0(00)
000	'memo'			1(01)
001	'copy'	111	'wire'	2(10)
000	'very'	011	'digi'	3(11)

3비트 32비트 3비트 32비트  
캐시(32바이트)

# Mapping: Set-Associative Mapping - H/W 조직



장점

- Fully Associative Mapping과 Direct Mapping 방식의 절충