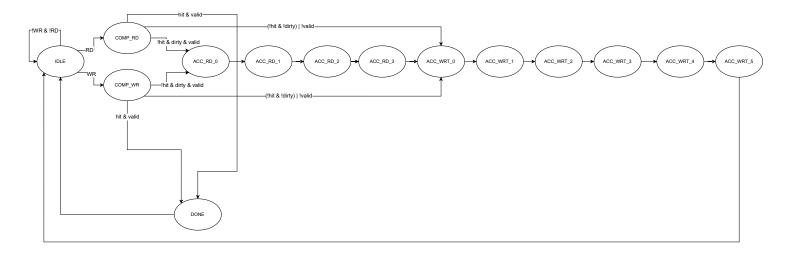
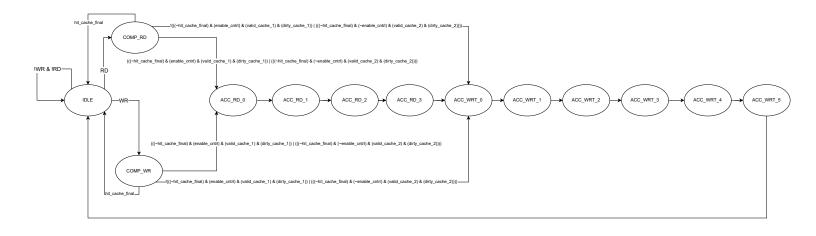
Direct-Mapped Cache State Diagram



State	Reset	IDLE	COMP_RD	COMP_WR	ACC_RD_0	ACC_RD_1	ACC_RD_2
Signal			_	_			
			valid cache	valid_cache?			
			_	(hit_cache?			
			? DONE :	DONE:			
		rd?	I	(dirty_cache			
		COMP_RD:	?	?			
	IDLE	(wr?		ACCESS_RD_	ACC RD 1	ACC_RD_2	ACC RD 3
		COMP_WR		0:			
		: IDLE)	ACCESS_W				
		,	R_0)):	_0)):			
				ACCESS_WR			
nxt_state			R_0	_0			
enable_cntrl	0	0	1	1	1	1	1
idx_cntrl	0	0	addr[10:3]	addr[10:3]	addr[10:3]	addr[10:3]	addr[10:3]
offset_cntrl	0	0	addr[2:0]	addr[2:0]	0	2	4
	0	0	1	1	0	0	0
comp_cntrl	l °	U	1	1	U	U	U
write_cntrl	0	0	0	1	0	0	0
tag_cntrl	х	X	addr[15:11]	addr[15:11]	Х	X	х
data_in_cntrl	х	х	х	data_in	х	х	х
valid_in_cntrl	0	0	0	0	0	1	2
addr_in_mem	х	х	х	х	х	х	х
	,	.,	· ·	.,	data_out_c	data_out_c	data_out_c
data_in_mem	х	х	х	Х	ache	ache	ache
write_mem	0	0	0	0	1	1	1
read_mem	0	0	0	0	0	0	0
Done	0	0	0	0	0	0	0
Stall	1	0	0	0	0	0	0
CacheHit	0	0	0	0	0	0	0
end_state	0	0	0	0	0	0	0
write	0	0	0	1	0	0	0
read	0	0	1	0	0	0	0
hit	0	0	0	1	0	0	0

ACC_RD_3	ACC_WR_0	ACC_WR_1	ACC_WR_2	ACC_WR_3	ACC_WR_4	ACC_WR_5	DONE
ACC_WR_0			ACC_WR_3	ACC_WR_4	ACC_WR_5	IDLE	IDLE
1	1	1	1	1	1	1	0
addr[10:3]	0	0	addr[10:3]	addr[10:3]	addr[10:3]	addr[10:3]	0
6	0	0	0	2	4	6	0
0	0	0	0	0	0	flop_write? 1'b1:1'b0	0
0	0	0	1	1	1	1	0
х	Х	х	addr[15:11]	addr[15:11]	addr[15:11]	addr[15:11]	х
x	х	х	(wr & (addr[2:0] == 3'b000)) ? data_in : data_out_me m	3'b010)) ? data_in :	(wr & (addr[2:0] == 3'b100)) ? data_in : data_out_me m	(flop_write & (addr[2:0] == 3'b110)) ? data_in : data_out_me m	х
3	0	0	1	1	1	1	0
х	{addr[15:3], 3'b000}	{addr[15:3], 3'b010}	{addr[15:3], 3'b100}	{addr[15:3], 3'b110}	х	х	х
data_out_ca che	х	х	х	х	х	х	х
1	0	0	0	0	0	0	0
0	1	1	1	1	0	0	0
0	0	0	0	0	0	1	1
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	flop_hit ? 1'b1 : 1'b0
0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Two-Way Associative Cache State Diagram



State	Reset	IDLE	COMP_RD	COMP_WR	ACC_RD_0	ACC_RD_1	ACC_RD_2	ACC_RD_3	ACC_WR_0	ACC_WR_1	ACC_WR_2	ACC_WR_3	ACC_WR_4	ACC_WR_5	
Signal															
		rd?	valid_cache ? (hit_cache ?	valid_cache ? (hit_cache ?											
	l	COMP_RD:	DONE : (dirty_cache ?	DONE : (dirty_cache ?											
	IDLE	(wr?	ACCESS_RD_0:	ACCESS_RD_0:	ACC_RD_1	ACC_RD_2	ACC_RD_3	ACC_WR_0	ACC_WR_1	ACC_WR_2	ACC_WR_3	ACC_WR_4	ACC_WR_5	IDLE	
		COMP_WR	ACCESS_WR_0)):	ACCESS_WR_0)):											
nxt_state	-	: IDLE)	ACCESS_WR_0	ACCESS_WR_0									$\overline{}$	\vdash	
			(valid_cache_1 &	(valid_cache_1 &											
			hit_cache_1)	hit_cache_1)											
			(~valid_cache_1 &	(~valid_cache_1 &											
			valid_cache_2 &	valid_cache_2 &											
			~hit_cache_final)	~hit_cache_final)											
	0	0	(~valid_cache_1 &	(~valid_cache_1 &	flop_en	flop_en	flop_en	flop_en	flop_en	flop_en	flop_en	flop_en	flop_en	flop_en	
			~valid_cache_2 &	~valid_cache_2 &								'			
			~hit_cache_final) (valid_cache_1 &	~hit_cache_final) (valid_cache_1 &											
			valid_cache_2 &	valid_cache_2 &											
			~hit_cache_final &	~hit_cache_final &											
anable entel			flop_victim_cntrl)	flop_victim_cntrl)											
enable_cntrl			(valid_cache_1 &	(valid_cache_1 &		<u> </u>	-		-				\vdash		
			hit_cache_1)	hit_cache_1)											
	I	I	(~valid_cache_1 &	(~valid_cache_1 &		1	1		l						
	I	I	valid_cache_1 &	valid_cache_1 &		1	1		l						
	I	I	~hit_cache_final)	~hit_cache_final)		1	1		l						
	Ι.	l .	(~valid_cache_1 &	(~valid_cache_1 &		1	1		l						
en	0	0	~valid_cache_2 &	~valid_cache_2 &											
		I	~hit_cache_final)	~hit_cache_final)		1	1		1						
			(valid_cache_1 &	(valid_cache_1 &											
		I	valid_cache_2 &	valid_cache_2 &		1	1		1						
			~hit_cache_final &	~hit_cache_final &											
			flop_victim_cntrl)	flop_victim_cntrl)											
idx_cntrl	0	0	addr[10:3]	addr[10:3]	addr[10:3]	addr[10:3]	addr[10:3]	addr[10:3]	0	0	addr[10:3]	addr[10:3]		addr[10:3]	
offset_cntrl	0	0	addr[2:0]	addr[2:0]	0	2	4	6	0	0	0	2	4	6	
comp_cntrl	0	0	0	1	0	0	0	0	0	0	0	0	0	flop_write? 1'b1:1'b0	
comp_rw	0	0	1	1	0	0	0	0	0	0	0	0	0	0	
							0	0	0	0	1	1		1	
[write_cntrt	1 0	0	1 0	1	1 0 1	1 0							l 1 l	1 1	
write_cntrl tag_cntrl	0 x	0 x	0 addr[15:11]		0 ×	0 x			x	x	_			addr[15:11]	
		0 x	0 addr[15:11]	1 addr[15:11]			x	×			addr[15:11] (wr &	addr[15:11] (wr &	addr[15:11] (wr &		
											addr[15:11]	addr[15:11]	addr[15:11]	addr[15:11]	
	х	х	addr[15:11]	addr[15:11]	×	х	х	х	х	х	addr[15:11] (wr & (addr[2:0] == 3'b000))?	addr[15:11] (wr & (addr[2:0] == 3'b010))?	addr[15:11] (wr & (addr[2:0] == 3'b100)) ?	addr[15:11] (flop_write & (addr[2:0] == 3'b110)) ?	
											addr[15:11] (wr & (addr[2:0] ==	addr[15:11] (wr & (addr[2:0] ==	addr[15:11] (wr & (addr[2:0] == 3'b100)) ? data_in:	addr[15:11] (flop_write & (addr[2:0] == 3'b110)) ? data_in :	
tag_cntrl	х	х	addr[15:11]	addr[15:11]	×	х	х	х	х	х	addr[15:11] (wr & (addr[2:0] == 3'b000))? data_in: data_out_me	addr[15:11] (wr & (addr[2:0] == 3'b010)) ? data_in : data_out_me	addr[15:11] (wr & (addr[2:0] == 3'b100)) ? data_in: data_out_me	addr[15:11] (flop_write & (addr[2:0] == 3'b110)) ? data_in: data_out_me	
	х	х	addr[15:11]	addr[15:11]	×	х	х	х	х	х	addr[15:11] (wr & (addr[2:0] == 3'b000)) ? data_in:	addr[15:11] (wr & (addr[2:0] == 3'b010))? data_in:	addr[15:11] (wr & (addr[2:0] == 3'b100)) ? data_in:	addr[15:11] (flop_write & (addr[2:0] == 3'b110)) ? data_in :	
tag_cntrl	х	х	addr[15:11]	addr[15:11]	×	х	х	х	х	х	addr[15:11] (wr & (addr[2:0] == 3'b000))? data_in: data_out_me	addr[15:11] (wr & (addr[2:0] == 3'b010)) ? data_in : data_out_me	addr[15:11] (wr & (addr[2:0] == 3'b100)) ? data_in: data_out_me	addr[15:11] (flop_write & (addr[2:0] == 3'b110)) ? data_in: data_out_me m	
tag_cntrl	х	х	addr[15:11]	addr[15:11]	×	х	х	х	х	х	addr[15:11] (wr & (addr[2:0] == 3'b000)) ? data_in : data_out_me m	addr[15:11] (wr & (addr[2:0] == 3'b010)) ? data_in : data_out_me m	addr[15:11] (wr & (addr[2:0] == 3'b100)) ? data_in : data_out_me m	addr[15:11] (flop_write & (addr[2:0] == 3'b110)) ? data_in : data_out_me m (flop_read &	
tag_cntrl	х	х	addr[15:11]	addr[15:11]	×	х	х	х	х	х	addr[15:11] (wr & (addr[2:0] == 3'b000))? data_in: data_out_me m	addr[15:11] (wr & (addr[2:0] == 3'b010)) ? data_in : data_out_me m (rd & (addr[2:0]	addr[15:11] (wr & (addr[2:0] == 3'b100))? data_in: data_out_me m (rd & (addr[2:0]	addr[15:11] (flop_write & (addr[2:0] == 3'b110)) ? data_in: data_out_me m	
tag_cntrl	x	х	addr[15:11]	addr[15:11]	×	х	х	х	х	х	addr[15:11] (wr & (addr[2:0] == 3'b000)) ? data_in : data_out_me m (rd & (addr[2:0] == 3'b000)) ?	addr[15:11] (wr & (addr[2:0] == 3'b010)) ? data_in : data_out_me m (rd & (addr[2:0] == 3'b010)) ?	addr[15:11] (wr & (addr[2:0] == 3'b100)) ? data_in: data_out_me m (rd & (addr[2:0] === 3'b100)) ?	addr[15:11] (flop_write & (addr[2:0] == 3'b110))? data_in: data_out_me m (flop_read & (addr[2:0] ===3'b110))?	
tag_cntrl data_in_cntrl	x	x	addr[15:11] x	addr[15:11] data_in	x	x	x	x	x	x	addr[15:11] (wr & (addr[2:0] == 3'b000)) ? data_in : data_out_me m (rd & (addr[2:0] == 3'b000)) ?	addr[15:11] (wr & (addr[2:0] == 3'b010)) ? data_in : data_out_me m (rd & (addr[2:0] ===3'b010))	addr[15:11] (wr & (addr[2:0] == 3'b100)) ? data_in: data_out_me m (rd & (addr[2:0] === 3'b100)) ?	addr[15:11] (flop_write & (addr[2:0] == 3'b110)) ? data_in : data_out_me m (flop_read & (addr[2:0] == 3'b110))	
tag_cntrl data_in_cntrl	x	x	addr[15:11] x	addr[15:11] data_in	x	x	x	x	x	x	addr[15:11] (wr & (addr[2:0] == 3'b000)) ? data_in : data_out_me m (rd & (addr[2:0] == 3'b000)) ? data_out_me m :	addr[15:11] (wr & (addr[2:0] == 3'b010)) ? data_in : data_out_me m (rd & (addr[2:0] == 3'b010)) ? data_out_me m :	addr[15:11] (wr & (addr[2:0] == 3'b100)); data_in: data_out_me m (rd & (addr[2:0] ===3'b100)) ? data_out_me m:	addr[15:11] (flop_write & (addr[2:0] == 3'b110)) ? data_in: data_out_me	
data_in_entrl	x x	x x 0	addr[15:11] x data_out_final	addr[15:11] data_in x	×	x	×	x	×	×	addr[15:11] (wr & (addr[2:0] == 3'b000))? data_in: data_out_me m (rd & (addr[2:0] ===3'b000)) ? data_out_me m: data_temp	addr[15:11] (wr & (addr[2:0] == 3'b010)); data_in: data_out_me m (rd & (addr[2:0] ===3'b010)) ; data_out_me m: data_temp	addr[15:11] (wr & (addr[2:0] == 3'b100))? data_in: data_out_me m (rd & (addr[2:0] ===3'b100)) ? data_out_me m: data_temp	addr[15:11] (flop_write & (addr[2:0] == 3'b110)) ? data_in : data_out_me	
tag_cntrl data_in_cntrl	x	x	addr[15:11] x	addr[15:11] data_in	x	x	x	x	x	x	addr[15:11] (wr & (addr[2:0] == 3'b000)) ? data_in : data_out_me m (rd & (addr[2:0] == 3'b000)) ? data_out_me m :	addr[15:11] (wr & (addr[2:0] == 3'b010)) ? data_in : data_out_me m (rd & (addr[2:0] == 3'b010)) ? data_out_me m :	addr[15:11] (wr & (addr[2:0] == 3'b100)); data_in: data_out_me m (rd & (addr[2:0] ===3'b100)) ? data_out_me m:	addr[15:11] (flop_write & (addr[2:0] == 3'b110)) ? data_in: data_out_me	
data_in_entrl	x x	x x 0	addr[15:11] x data_out_final	addr[15:11] data_in x	x x	x x x	x x x	x x x	x x x	x x x	addr[15:11] (wr & (addr[2:0] == 3'b000)) ? data_in: data_out_me m (rd & (addr[2:0] == 3'b000)) ? data_out_me m: data_temp 1	addr[15:11] (wr & (addr[2:0] == 3'b010)) ? data_in: data_out_me m (rd & (addr[2:0] == 3'b010)) ? data_out_me m: data_cut_me m: data_temp 1	addr[15:11] (wr & (addr[2:0] == 3'b100))? data_in: data_out_me m (rd & (addr[2:0] ===3'b100)) ? data_out_me m: data_temp	addr[15:11] (flop_write & (addr[2:0] == 3'b110)) ? data_in : data_out_me	
data_in_entrl	x x 0	x x 0	addr[15:11] x data_out_final	addr[15:11] data_in x	x x x 4	x x x 0 (tag.out.final	x x x 0 ttag_out_final	x x x 4	x x x 0 0 {addr(15:3),	x x 0 (addr[15:3],	addr[15:11] (wr & (addr[2:0] = 3'b000); data_in: data_out_me m (rd & (addr[2:0] == 3'b000); ? data_out_me m: data_temp 1 (addr[15:3],	addr[15:11] (wr & (addr[2:0] == 3'b010)); data_in: data_out_me m (rd & (addr[2:0] == 3'b010)); ? data_out_me m: data_temp 1 [addr[15:3],	addr[15:11] (wr & (addr[2:0] == 3*1001); data_in: data_out_me m (rd & (addr[2:0] === 3*1001); ? data_out_me m: data_temp 1	addr[15:11] (flop_write & (addr[2:0] == 37b110))? data_in: data_out_me m (flop_read & (addr[2:0] === 37b110)) ? data_out_me m: data_temp 1	
data_in_entrl	x x	x x 0	addr[15:11] x data_out_final	addr[15:11] data_in x	x x x (1) (tag_out_final_l_idx_cnttf,	x x x (tag_out_final, idx_entri,	x x x (0) (tag_out_final, idx_cntrl, idx_cnt	x x x 4 4tag_out_fina	x x x	x x x	addr[15:11] (wr & (addr[2:0] == 3'b000)) ? data_in: data_out_me m (rd & (addr[2:0] == 3'b000)) ? data_out_me m: data_temp 1	addr[15:11] (wr & (addr[2:0] == 3'b010)) ? data_in: data_out_me m (rd & (addr[2:0] == 3'b010)) ? data_out_me m: data_cut_me m: data_temp 1	addr[15:11] (wr & (addr[2:0] == 3'b100))? data_in: data_out_me m (rd & (addr[2:0] ===3'b100)) ? data_out_me m: data_temp	addr[15:11] (flop_write & (addr[2:0] == 3'b110)) ? data_in : data_out_me	
data_in_entrl	x x 0	x x 0	addr[15:11] x data_out_final	addr[15:11] data_in x	x x x 4	x x x 0 (tag.out.final	x x x 0 ttag_out_final	x x x 4 4tag_out_fina	x x x 0 0 {addr(15:3),	x x 0 (addr[15:3],	addr[15:11] (wr & (addr[2:0] = 3'b000); data_in: data_out_me m (rd & (addr[2:0] == 3'b000); ? data_out_me m: data_temp 1 (addr[15:3],	addr[15:11] (wr & (addr[2:0] == 3'b010)); data_in: data_out_me m (rd & (addr[2:0] == 3'b010)); ? data_out_me m: data_temp 1 [addr[15:3],	addr[15:11] (wr & (addr[2:0] == 3*1001); data_in: data_out_me m (rd & (addr[2:0] === 3*1001); ? data_out_me m: data_temp 1	addr[15:11] (flop_write & (addr[2:0] == 37b110))? data_in: data_out_me m (flop_read & (addr[2:0] === 37b110)) ? data_out_me m: data_temp 1	
data_in_entrl data_out_entrl valid_in_entrl addr_in_mem	0 0 x	0 0 x	addr[15:11] x data_out_final 0	addr[15:11] data_in x 0	x x (tag_out_fina, idx_cntrl, offset_cntrl) data_out_fin	x x x (tag_out_final, idx_entri,	x x x 0 (tag_out_final_idx_cntrl, offset_cntrl) data_out_fin	x x x Q Q (tag_out_fina l_idx_cntrl, offset_cntrl) data_out_fin	x x x 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4	x x 4 4 4 4 5 4 5 5 6 7 7 8 7 8 7 8 7 8 7 8 8 8 8 8 8 8 8 8	addr[15:11] (wr & (addr[2-0] = 3'b000))? data_in: data_out_me m (rd & (addr[2:0] === 3'b000))? data_out_me m: data_temp 1 {addr[15:3], 3'b100}	addr[15:11] (wr & (addr[2:0] = 3'b010))? data_in: data_out_me m (rd & (addr[2:0] == 3'b010))? data_in: data_out_me in: data_temp 1 (addr[15:3], 3'b110}	addr[15:11] (wr & (addr[20] = 3'b100))? data_in: data_out_me m (rd & (addr[2:0] ===3'b100)) ? data_out_me m: data_temp 1	addr[15:11] (flop_write & (addr[2.0] = 31110))	
data_in_cntrl data_out_cntrl valid_in_cntrl addr_in_mem data_in_mem	0 0 x	0 0 x	addr[15:11] x data_out_final 0 x	addr[15:11] data_in x 0 x	x x (tag_out_fina i, idx_cntrl, offset_cntrl) data_out_fin al	x x (tag_out_final, idx_cntrl, offset_cntrl) data_out_final	x x (tag_out_final, idx_cntrl, offset_cntrl) data_out_final	x x (tag_out_fina , idx_cntrl, offset_cntrl) data_out_fin al	x x x 4 4 4 4 4 4 5 4 5 6 7 7 8 7 8 7 8 8 8 8 8 8 8 8 8 8 8 8 8	x x x 0 {addr[15:3], 3'b010}	addr[15:11] (wr & (addr[2-0] = 3'b000)) ? data_in: data_out_me m (rd & (addr[2:0] === 3'b000)) ? data_out_me n: data_temp 1 {addr[15:3], 3'b100}	addr[15:11] (wr & (addr[2:0] = 3'b010))? data_in: data_out_me m (rd & (addr[2:0] == 3'b010)) ? data_out_me m: data_temp 1 {addr[15:3], 3'b110}	addr[15:11] (wr & (addr[20] = 3'b100))? data_in: data_out_me m (rd & (addr[2:0] ===3'b100)) ? data_out_me m: data_temp 1 x	addr[15:11] (flop_write & (addr[2:0] = 31110))	
data_in_cntrl data_out_cntrl valid_in_cntrl addr_in_mem data_in_mem write_mem	0 0 x x 0 0	0 0 x x 0 0	addr[15:11] x data_out_final 0 x x	addr[15:11] data_in x 0 x x 0	x x (a) (tag_out_fina i, idx_cntrl, offset_cntrl) (data_out_fin al	x x (tag_out_final, idx_cntrl, offset_cntrl) data_out_final, idx_cntrl, offset_cntrl)	x x 4 4 4 4 4 4 4 4 4 4 4 4	x x x 4tag_out_fina l, idx_cntrl, offset_cntrl) data_out_fin al	x x x 0 {addr[15:3], 3'b000} x 0	x x x 0 {addr[15:3], 3'b010} x 0	addr[15:11] (wr & (addr[2-0] (addr[1-0] (add	addr[15:11] (wr & (addr[2:0] = 3'b010))? data_in: data_out_me m (rd & (addr[2:0] ==-3'b010))? ? data_out_me m: data_temp 1 {addr[15:3], 3'b110} x	addr[15:11] (wr & (addr[20] = 3'b100))? data_in: data_out_me m (rd & (addr[20] = 3'b100))? data_out_me m: data_out_me n: data_out_me x x 0	addr[15:11] (flop_write & (addr[2:0] = 3*b110))	
data_in_entrl data_out_entrl valid_in_entrl addr_in_mem data_in_mem	0 0 x	0 0 x	addr[15:11] x data_out_final 0 x	addr[15:11] data_in x 0 x	x x (tag_out_fina i, idx_cntrl, offset_cntrl) data_out_fin al	x x (tag_out_final, idx_cntrl, offset_cntrl) data_out_final	x x (tag_out_final, idx_cntrl, offset_cntrl) data_out_final	x x (tag_out_fina , idx_cntrl, offset_cntrl) data_out_fin al	x x x 4 4 4 4 4 4 5 4 5 6 7 7 8 7 8 7 8 8 8 8 8 8 8 8 8 8 8 8 8	x x x 0 {addr[15:3], 3'b010}	addr[15:11] (wr & (addr[2-0] = 3'b000)) ? data_in: data_out_me m (rd & (addr[2:0] === 3'b000)) ? data_out_me n: data_temp 1 {addr[15:3], 3'b100}	addr[15:11] (wr & (addr[2:0] = 3'b010))? data_in: data_out_me m (rd & (addr[2:0] == 3'b010)) ? data_out_me m: data_temp 1 {addr[15:3], 3'b110}	addr[15:11] (wr & (addr[20] = 3'b100))? data_in: data_out_me m (rd & (addr[2:0] ===3'b100)) ? data_out_me m: data_temp 1 x	addr[15:11] (flop_write & (addr[2:0] = 3'b110)) ? data_in: data_out_me	
data_in_cntrl data_out_cntrl valid_in_cntrl addr_in_mem data_in_mem write_mem	0 0 x x 0 0	0 0 x x 0 0	addr[15:11] x data_out_final 0 x x	addr[15:11] data_in x 0 x x 0	x x (a) (tag_out_fina i, idx_cntrl, offset_cntrl) (data_out_fin al	x x (tag_out_final, idx_cntrl, offset_cntrl) data_out_final, idx_cntrl, offset_cntrl)	x x 4 4 4 4 4 4 4 4 4 4 4 4	x x x 4tag_out_fina l, idx_cntrl, offset_cntrl) data_out_fin al	x x x 0 {addr[15:3], 3'b000} x 0	x x x 0 {addr[15:3], 3'b010} x 0	addr[15:11] (wr & (addr[2-0] (addr[1-0] (add	addr[15:11] (wr & (addr[2:0] = 3'b010))? data_in: data_out_me m (rd & (addr[2:0] ==-3'b010))? ? data_out_me m: data_temp 1 {addr[15:3], 3'b110} x	addr[15:11] (wr & (addr[20] = 3'b100))? data_in: data_out_me m (rd & (addr[20] = 3'b100))? data_out_me m: data_out_me n: data_out_me x x 0	addr[15:11] (flop_write & (addr[2:0] = 35110))	
data_in_entrl data_out_entrl valid_in_entrl addr_in_mem data_in_mem write_mem read_mem	x x 0 0 x x	x x 0 0 x x	addr[15:11] x data_out_final 0 x x x 0 0 x	addr[15:11] data_in x 0 x x 0 x x 0 x	x x 4 4 4 4 4 4 4 4 4 4 4 4	x x Q (tag_out_final, idx_entri, offset_entri) data_out_final 1 0 x	x x 4tag_out_final ,idx_cntrl, offset_cntrl) data_out_fin al 1 0 x	x x 4 tag_out_fina l, idx_cntrl, offset_cntrl) data_out_fin al 1 0 x	x x 0 {addr[15:3], 3'b000} x 0 1 x	x x 0 (addf[15:3], 3'b010) x 0 1 x	addr[15:11] (wr & (addr[2-1]= 3'b000)) ? data_in: data_out_me m (rd & (addr[2:0] == 3'b000) ? data_out_me m: data_temp 1 1 (addr[15:3], 3'b100) x	addr[15:11] (wr & (addr[2:0] = 3'b010))? data_in: data_out_me m (rd & (addr[2:0] === 3'b010))? data_out_me m: data_temp 1 (addr[15:3], 3'b110) x 0 1	addr[15:11] (wr & (addr[20] = 3'b100))? data_in: data_out_me (rd & (addr[20] = 3'b100))? data_out_me m: data_out_me m: data_temp 1 x x 0 0 0	addr[15:11] (flop_write & (addr[2.0] = 3'b110)) ? data_in' data_out_me m (flop_read & (addr[2.0] == 3'b110) ? data_tot_me n: data_temp 1 x x x 0 0 -flop_victim _cntrl	
data_in_cntrl data_out_cntrl valid_in_cntrl addr_in_mem data_in_mem write_mem read_mem victim_cntrl	x x 0 0 0	x x 0 0 0	addr[15:11] x data_out_final 0 x x 0 0	addr[15:11] data_in x 0 x x 0 0 0	x x (tag_out_fina l, idx_cntrl, offset_cntrl) data_out_fin al 1 0	x x 0 (tag_out_final, idx_cntrl, offset_cntrl) data_out_fina l 1 0	x x 4 4 4 4 4 4 4 4 4 4 4 4	x x (tag_out_fina i, idx_cntrl, offset_cntrl) (data_out_fin al 1 0	x x 4addr[15:3], 3'b000} x 0 1	x x 4 4 4 4 4 4 4 4 4 4 4 4	addr[15:11] (wr & (addr[2-0] (addr[15:3], 3'b100) x 0 1	addr[15:11] (wr & (addr[2:0] = 3'b010))? data_in: data_out_me m (rd & (addr[2:0] ==-3'b010)) ? data_out_me m: data_temp 1 1 (addr[15:3], 3'b110} x 0 1	addr[15:11] (wr & (addr[20] = 3°b100))? data_in: data_out_me (rd & (addr[20] = 3°b100))? data_out_me m: (addr[20] data_out_me x x 0 0	addr[15:11] (flop_write & (addr[2:0] = 35110))	
data_in_entrl data_out_entrl valid_in_entrl addr_in_mem data_in_mem write_mem read_mem victim_entrl Done	x x 0 0 x 0 x 0 0 x	x x 0 0 x 0 x 0 0 x	addr[15:11] x data_out_final 0 x x 0 0 0 x hit_cache_final	addr[15:11] data_in x 0 x 0 x 0 0 x	x x x (tag_out_fina _idx_cntrl, offset_cntrl) data_out_fin al 1 0 x 0	x x (tag_out_final ,idx_cntrl, offset_cntrl) data_out_fina l 1 0 x	x x (tag_out_final , idx_cntrl, offset_cntrl) data_out_fin al 1 0 x 0	x x x (tag_out_fina lidx_cntrl, offset_cntrl) data_out_fin al 1 0 x 0	x x x 4 4 4 4 4 5 4 7 7 8 7 8 7 8 7 8 8 8 8 8 8 8 8 8 8 8	x x x 0 {addr[15:3], 3'b010} x 0 1 x 0	addr[15:11] (wr & (addr[2:0] = 3'b000))? data_in: data_out_me m (rd & (addr[2:0] ===3'b000)) ? data_out_me m: data_temp 1 {addr[15:3], 3'b100} x 0 1 x	addr[15:11] (wr & (addr[2:0] = 3'b010))? data_out_me m (rd & (addr[2:0] === 3'b010))? data_out_me m: (addr_2:0] === 3'b010) ? data_out_me 1 4addr[15:3], 3'b110} x 0 1 x	addr[15:11] (wr & (addr[20] = 3'b100))? data_in: data_out_me m (rd & (addr[2:0] === 3'b100))? data_in: data_temp 1 x 0 0 x 0	addr[15:11] (flop_write & (addr[2:0] = 3*b110))	
data_out_entrl data_out_entrl addr_in_mem data_in_mem write_mem red_mem victim_entrl Done Stalt CacheHit end_state	x x 0 0 x x 0 0 x 0 1 0 0 0 0 0 0 0 0 0	x x 0 0 0 x x 0 0 0 0 0 0 0 0 0 0 0 0 0	addr[15:11] x data_out_final 0 x x 0 0 x hit_cache_final 0	addr[15:11] data_in x 0 x 0 x 0 0 x 0 0 0	x x (tag_out_fina	x x (tag_out_final ,idx_cntrl, offset_cntrl) data_out_fina t 1 0 x 0 0 0	x x (tag_out_final , idx_cntrl, offset_cntrl) data_out_fin al 1 0 x 0 0 0	x x x (tag_out_final line line line line line line line lin	x x x 4 4 4 4 5 4 6 6 7 7 8 7 8 7 8 7 8 8 8 8 8 8 8 8 8 8	x x x 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4	addr[15:11] (wr & (addr[2:0] = 3'b000))? data_in: data_out_me m (rd & (addr[2:0] ===3'b000)) ? data_out_me m: :data_temp 1 {addr[15:3], 3'b100} x 0 1 x 0 0 0 0	addr[15:11] (wr & (addr[2:0] = 3'b010))? data_un': data_un': data_un': (rd & (addr[2:0] === 3'b010))? data_ut_me m: :data_temp 1 (addr[15:3], 3'b110) x 0 1 x 0 0 0 0	addr[15:11] (wr & (addr[20] = 3'b100))? data_in: data_out_me m (rd & (addr[2:0] === 3'b100))? data_in: data_temp 1 x x 0 0 x 0 0 0 0	addr[15:11] (flop_write & (addr[2:0] = 3°b110)) ? data_in: data_out_me (flop_read & (addr[2:0] = -3°b110)) ? ? data_out_me m: data_temp 1	
data_in_cntrl data_out_cntrl valid_in_cntrl addr_in_mem write_mem read_mem victim_cntrl Done Stall CacheHit end_state write	x x 0 0 x x 0 0 x 0 1 0 0 0 0 0 0 0 0 0 0 0	x x x 0 0 x x 0 0 0 x 0 0 0 0 0 0 0 0	addr[15:11] x data_out_final 0 x x 0 0 x hit_cache_final hit_cache_final	addr[15:11] data_in x 0 x x 0 0 x 0 0 0 1	x x (tag_out_fina l, idx_entrl, offset_entrl) data_out_fin al 1 0 x 0 0 0 0	x x (tag_out_final, idx_cntrl, offset_cntrl) data_out_final, idx_cntrl, offset_cntrl) data_out_final, idx_cntrl, offset_cntrl) and offset_cntrl) and offset_cntrl)	x x x 4tag_out_final, idx_cntrl, offset_cntrl) data_out_fin al 1 0 x 0 0 0 0	x x (tag_out_fina), idx_cntrl, offset_cntrl) data_out_fin 1 0 x 0 0 0 0	x x 4addr[15:3], 3'b000} x 0 1 x 0 0 0 0	x x 0 {addr[15:3], 3'b010} x 0 1 x 0 0 0 0	addr[15:11] (wr & (addr[2-1] (abdr[2-1] (abdr[2-1] (abdr[2-1] (abdr[2-1] (abdr[2-1] (abdr[2-1] (abdr[2-1] (abdr[2-1] (abdr[15:3] (abdr[15:	addr[15:11] (wr & (addr[2:0] = 3'b010))? data_in: data_out_me m (rd & (addr[2:0] ==-3'b010))? ? data_out_me m: data_temp 1 1 (addr[15:3], 3'b110} x 0 1 x 0 0 0 0	addr[15:11] (wr & (addr[20] = 3'b100))? data_in: data_out_me (rd & (addr[20] = 3'b100))? data_in: data_out_me m: data_out_me x x 0 0 0 0 0 0	addr[15:11] (flop_write & (addr[2:0] = 3*b110)) ? data_in: data_out_me m (flop_read & (addr[2:0] == 3*b110)) ? data_out_me m: data_temp 1 x x	
data_in_entrl data_out_entrl data_out_entrl addr_in_entrl addr_in_mem write_mem read_mem victim_entrl Done Stall CacheHit end_state	x x 0 0 x x 0 0 x 0 1 0 0 0 0 0 0 0 0 0	x x 0 0 0 x 0 0 0 0 0 0 0 0 0 0 0 0 0 0	addr[15:11] x data_out_final 0 x x 0 0 0 x hit_cache_final ht_cache_final	addr[15:11] data_in	x x (tag_out_fina	x x (tag_out_final ,idx_cntrl, offset_cntrl) data_out_fina t 1 0 x 0 0 0	x x (tag_out_final , idx_cntrl, offset_cntrl) data_out_fin al 1 0 x 0 0 0	x x x (tag_out_final line line line line line line line lin	x x x 4 4 4 4 5 4 6 6 7 7 8 7 8 7 8 7 8 8 8 8 8 8 8 8 8 8	x x x 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4	addr[15:11] (wr & (addr[2:0] = 3'b000))? data_in: data_out_me m (rd & (addr[2:0] ===3'b000)) ? data_out_me m: :data_temp 1 {addr[15:3], 3'b100} x 0 1 x 0 0 0 0	addr[15:11] (wr & (addr[2:0] = 3'b010))? data_un': data_un': data_un': (rd & (addr[2:0] === 3'b010))? data_ut_me m: :data_temp 1 (addr[15:3], 3'b110) x 0 1 x 0 0 0 0	addr[15:11] (wr & (addr[20] = 3'b100))? data_in: data_out_me m (rd & (addr[2:0] === 3'b100))? data_in: data_temp 1 x x 0 0 x 0 0 0 0	addr[15:11] (flop_write & (addr[2:0] = 31:110))	