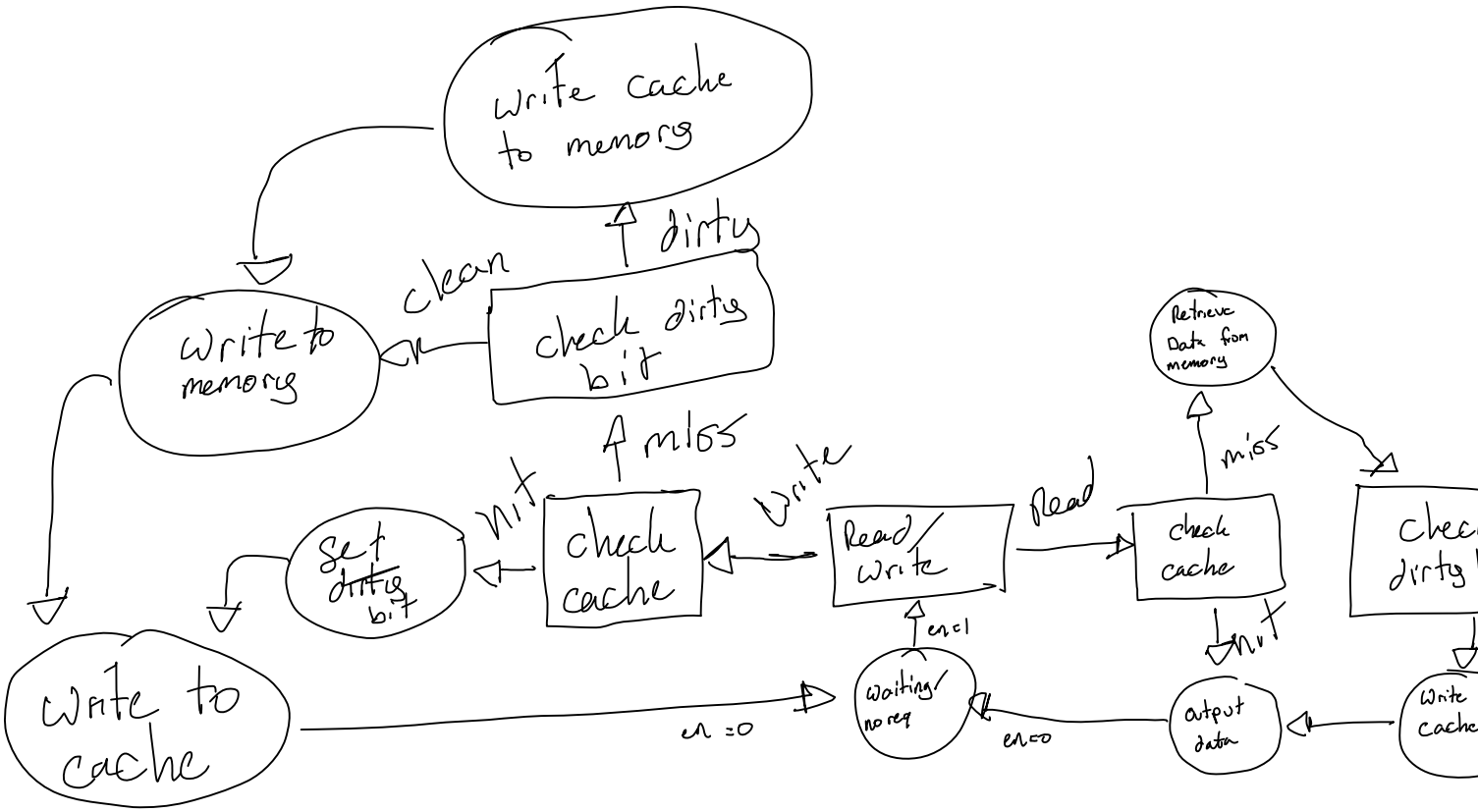
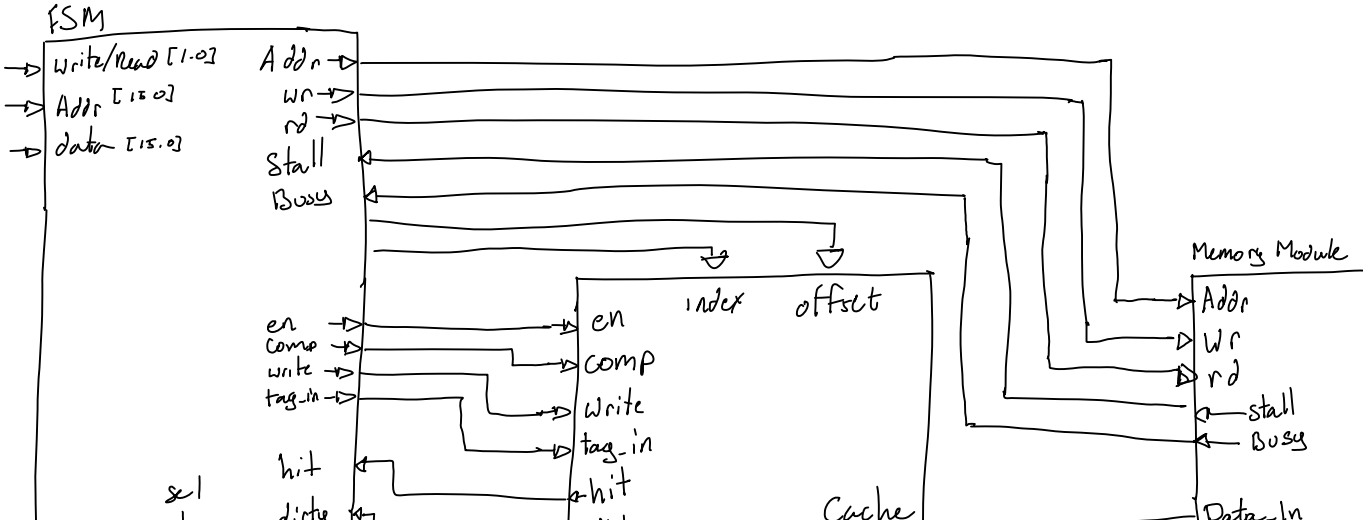
Direct mapped cache FSM

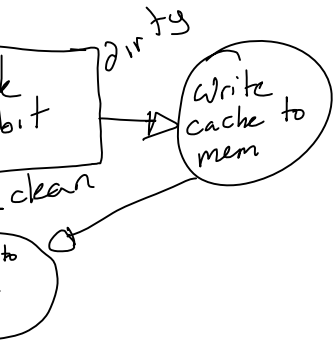


Direct Mapped Cache Schematic



check cache
- Addr in (Tag, Index, Offset)
- comp = 1
  - Read → write=0
  - write → write = 1
- check hit bit output from cache

check/set dirty bit
- Addr in (Tag, Index, Offset)
- set automatically by cache module

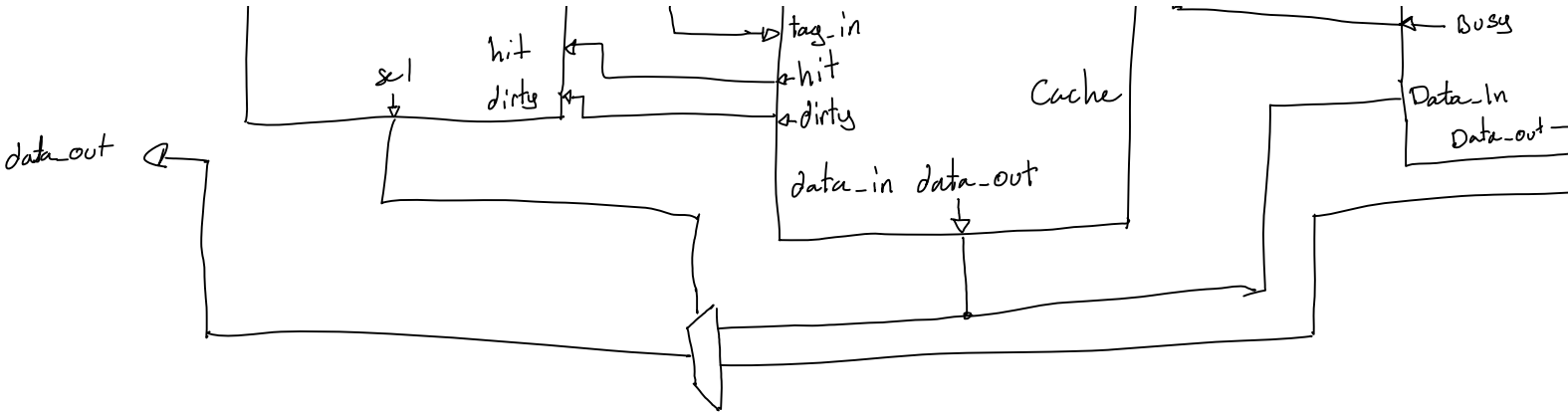


Cache I/O

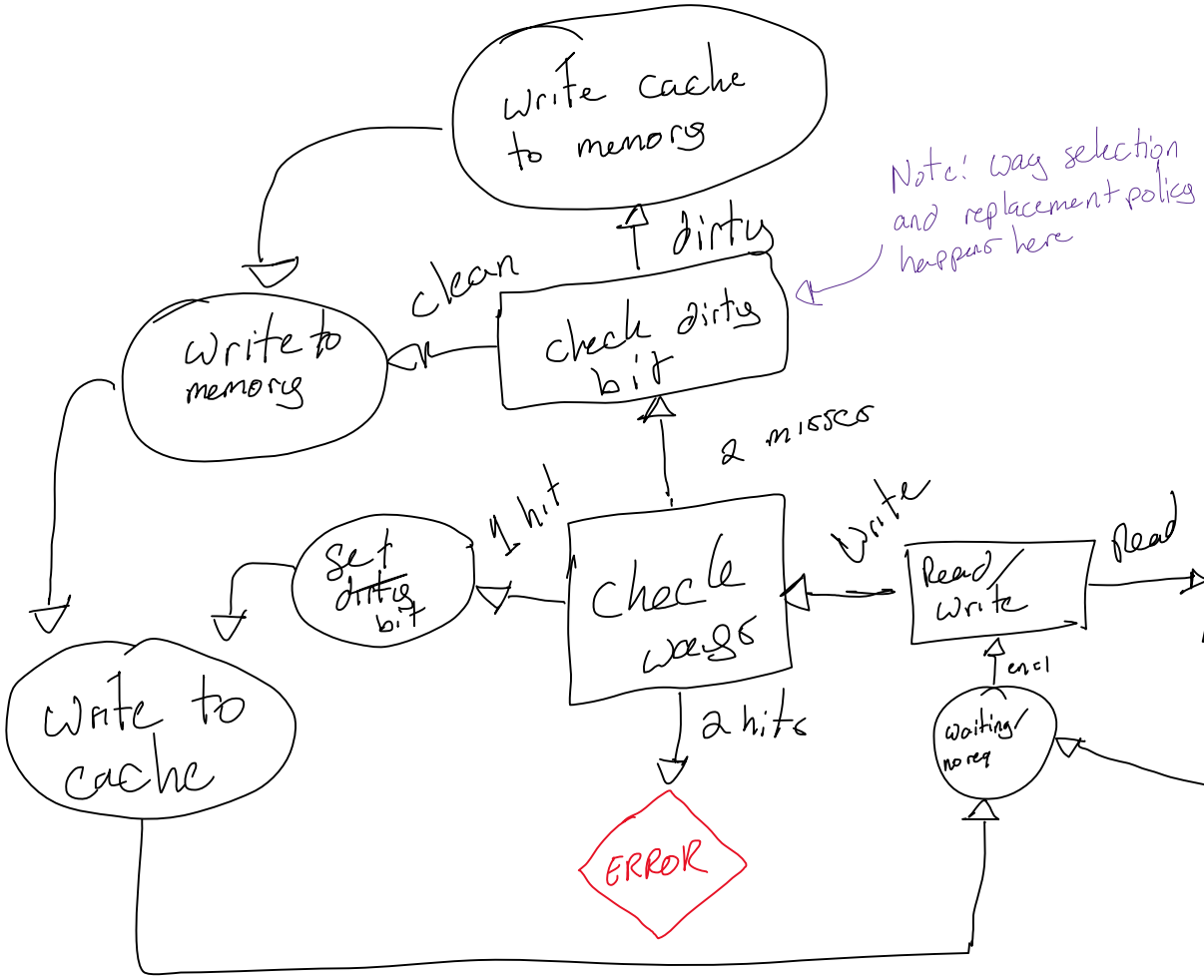
FSM →	enable	In	1	Enable cache. Active high. If low, "write" and "comp" have no effect, and all outputs are zero.
	index	In	8	The address bits used to index into the cache memory.
	offset	In	3	offset[2:1] selects which word to access in the cache line. The least significant bit should be 0 for word alignment. If the least significant bit is 1, it is an error condition.
FSM →	comp	In	1	Compare. When "comp"=1, the cache will compare tag_in to the tag of the selected line and indicate if a hit has occurred; the data portion of the cache is read or written but writes are suppressed if there is a miss. When "comp"=0, no compare is done and the Tag and Data portions of the cache will both be read or written.
FSM →	write	In	1	Write signal. If high at the rising edge of the clock, a write is performed to the data selected by "index" and "offset", and (if "comp"=0) to the tag selected by "index".

Memory Module

	Signal	In/Out	Width	Description
FSM →	Addr	In	16	Provides the address to perform an operation on.
Cache →	DataIn	In	16	Data to be used on a write.
FSM →	wr	In	1	When wr="1", the data on DataIn will be written to Mem[Addr] four cycles after wr is asserted.
FSM →	rd	In	1	When rd="1", the DataOut will show the value of Mem[Addr] two cycles after rd is asserted.
	clk	In	1	Clock signal; rising edge active.
	rst	In	1	Reset signal. When "rst"=1, the memory will load the data from the file "loadfile".
	createdump	In	1	Write contents of memory to file. Each bank will be written to a different file, named dumpfile_[0-3].

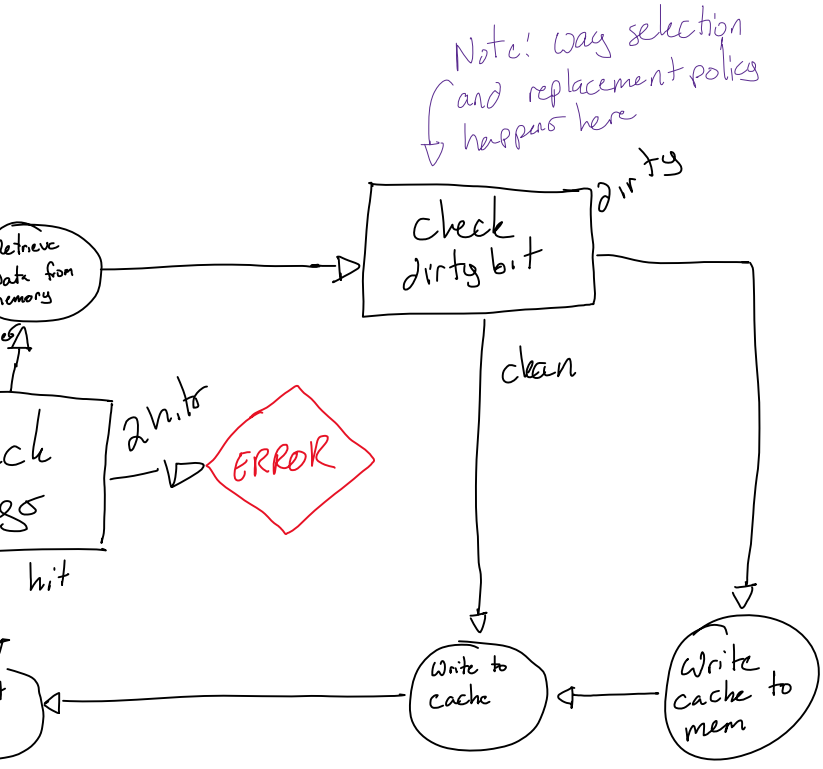


2-Way Set Associative Cache FSM



				if a hit has occurred; the data portion of the cache is read or written but writes are suppressed if there is a miss. When "comp"=0, no compare is done and the Tag and Data portions of the cache will both be read or written.
FSM →	write	In	1	Write signal. If high at the rising edge of the clock, a write is performed to the data selected by "index" and "offset", and (if "comp"=0) to the tag selected by "index".
FSM →	tag_in	In	5	When "comp"=1, this field is compared against stored tags to see if a hit occurred; when "comp"=0 and "write"=1 this field is written into the tag portion of the array.
MEM →	data_in	In	16	On a write, the data that is to be written to the location specified by the "index" and "offset" inputs.
FSM →	valid_in	In	1	On a write when "comp"=0, the data that is to be written to valid bit at the location specified by the "index" input.
	clk	In	1	Clock signal; rising edge active.
	rst	In	1	Reset signal. When "rst"=1 on the rising edge of the clock, all lines are marked invalid. (The rest of the cache state is not initialized and may contain X's.)
	createdump	In	1	Write contents of entire cache to memory file. Active on rising edge.
FSM ←	hit	Out	1	Goes high during a compare if the tag at the location specified by the "index" lines matches the "tag_in" lines.
FSM ←	dirty	Out	1	When this bit is read, it indicates whether this cache line has been written to. It is valid on a read cycle, and also on a compare-write cycle when hit is false. On a write with "comp"=1, the cache sets the dirty bit to 1. On a write with "comp"=0, the dirty bit is reset to 0.
	tag_out	Out	5	When "write"=0, the tag selected by "index" appears on this output. (This value is needed during a writeback.)
FSM ← MEM ←	data_out	Out	16	When "write"=0, the data selected by "index" and "offset" appears on this output.
	valid	Out	1	During a read, this output indicates the state of the valid bit in the selected cache line.

FSM →	rd	In	1	When rd="1", the DataOut will show the value of Mem[Addr] two cycles after rd is asserted.
	clk	In	1	Clock signal; rising edge active.
	rst	In	1	Reset signal. When "rst"=1, the memory will load the data from the file "loadfile".
	createdump	In	1	Write contents of memory to file. Each bank will be written to a different file, named dumpfile_[0-3]. Active on rising edge.
Cache →	DataOut	Out	16	Two cycles after rd="1", the data at Mem[Addr] will be shown here.
FSM ←	stall	Out	1	Is set to high when the operation requested at the input cannot be completed because the required bank is busy.
	Busy	Out	4	Shows the current status of each bank. High means the bank cannot be accessed.
	err	Out	1	The error signal is raised on an unaligned access.



Way 1
h
h
m
m

Write offset 0 → Write offset 1

way 2

n → Nope, not possible, error

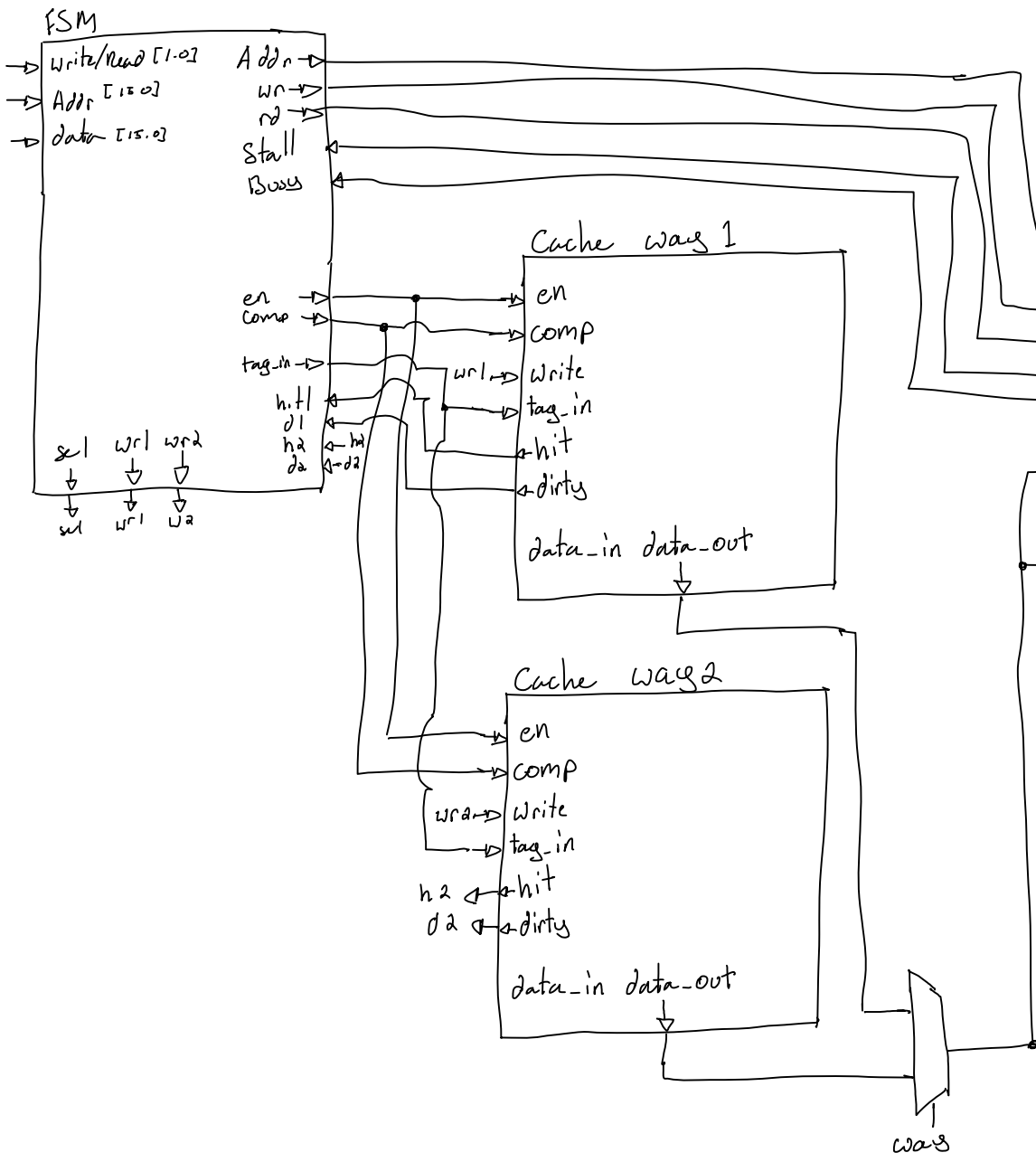
m } normal hit

h

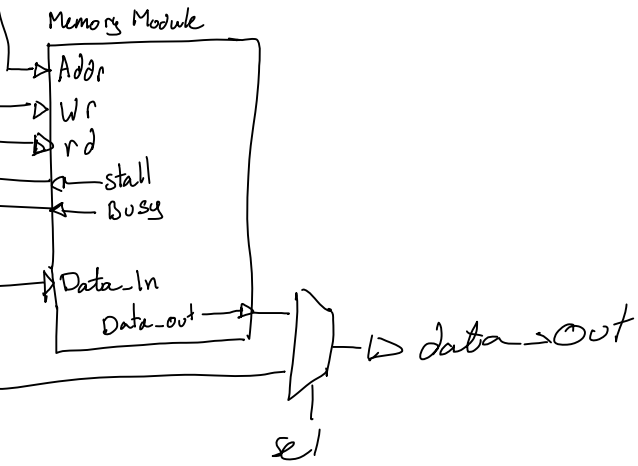
m → miss, pick a cache to overwrite

→ write offset 2 → write offset 3

2 way set associative cache schematic



Write offset 0 → Write offset 1



→ write → offset 2 → offset 3