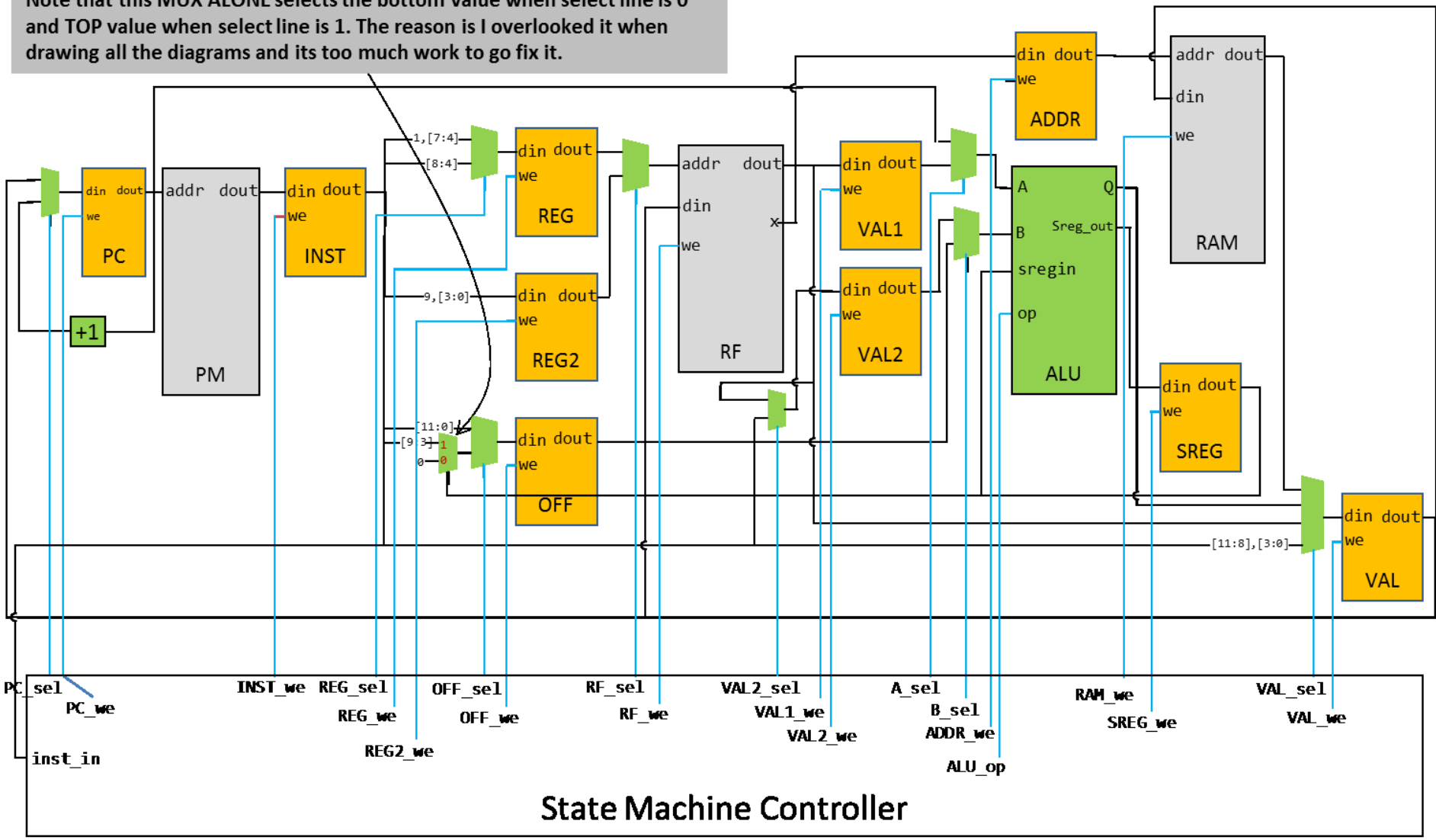
Today

- Today: Sequential elements
  - 1-bit memory cell
  - 8-bit auxilliary register
  - Register files and memories

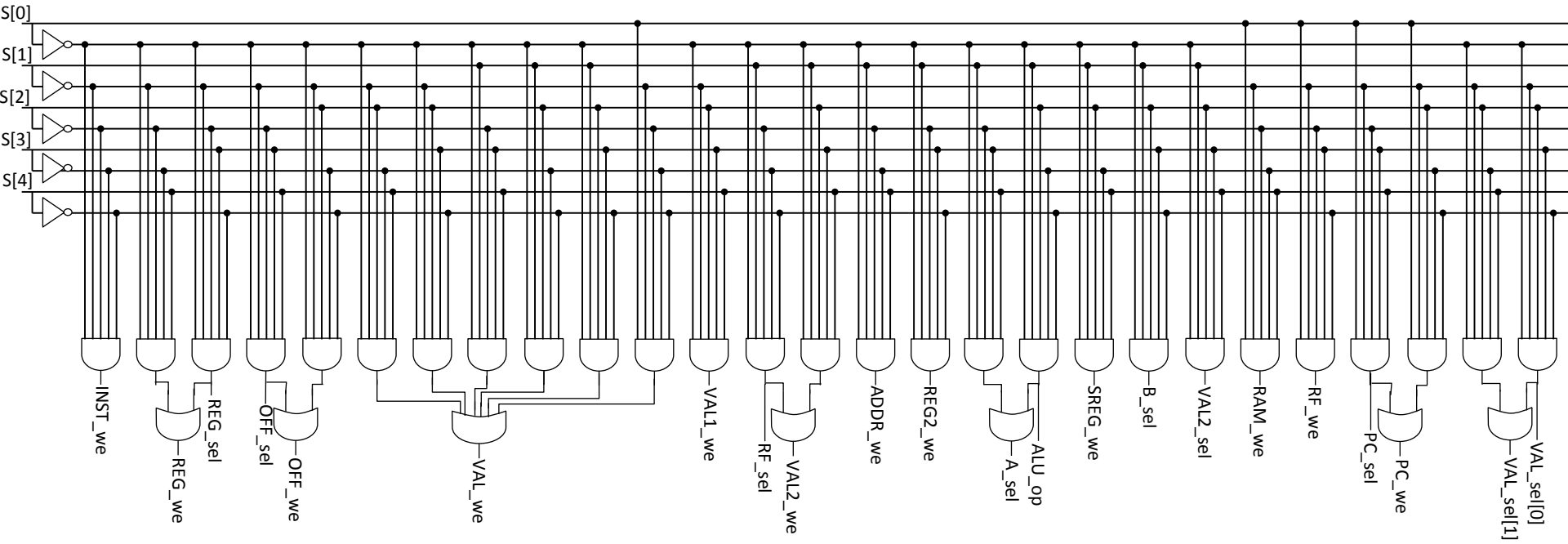
State machine controller

Note that this MUX ALONE selects the bottom value when select line is 0 and TOP value when select line is 1. The reason is I overlooked it when drawing all the diagrams and its too much work to go fix it.

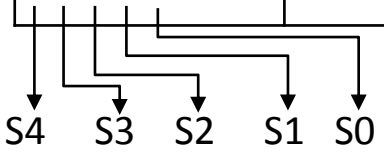


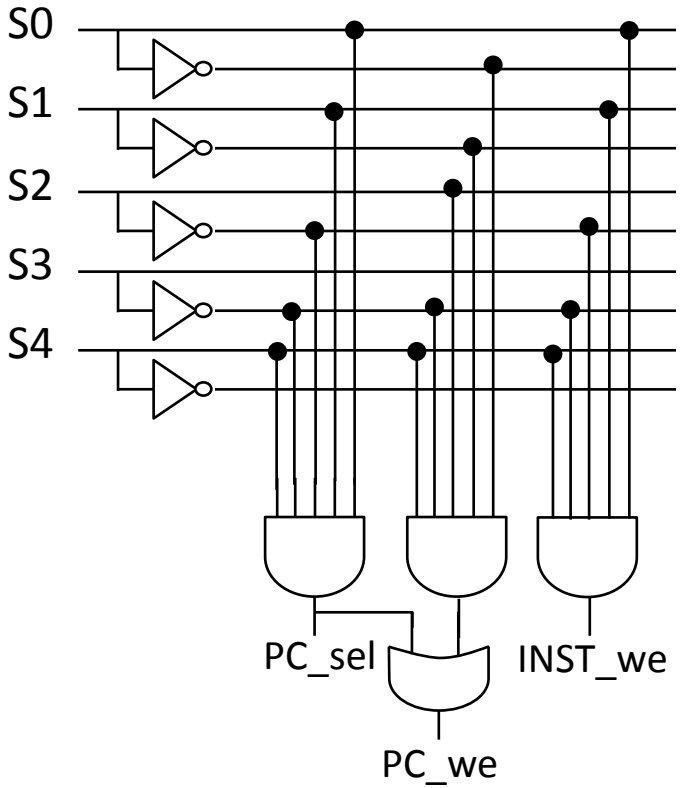
State number	INST_we	REG_we	REG_sel	OFF_sel	OFF_we	VAL_we	VAL1_we	VAL2_we	RF_sel	ADDR_we	REG2_we	A_sel	ALU_op	SREG_we	B_sel	VAL2_sel	RAM_we	RF_we	PC_sel	PC_we	VAL_sel1	VAL_sel0
00000	1	0			0	0	0	0		0	0			0			0	0		0		
00001	0	1	0		0	0	0	0		0	0			0			0	0		0		
00010	0	1	1		0	0	0	0		0	0			0			0	0		0		
00011	0	0		1	1	0	0	0		0	0			0			0	0		0		
00100	0	0		0	1	0	0	0		0	0			0			0	0		0		
00101	0	0			0	1	0	0		0	0			0			0	0		0	1	1
00110	0	0			0	1	0	0	0	0	0			0			0	0		0	1	0
00111	0	0			0	0	1	0	0	0	0			0			0	0		0		
01000	0	0			0	0	0	1	1	0	0			0		0	0	0		0		
01001	0	0			0	0	0	0		1	0			0			0	0		0		
01010	0	0			0	0	0	0		0	1			0			0	0		0		
01011	0	0			0	1	0	0		0	0	1	0	0	0		0	0		0	0	0
01100	0	0			0	1	0	0		0	0	1	1	0	0		0	0		0	0	0
01101	0	0			0	0	0	0		0	0			1			0	0		0		
01110	0	0			0	1	0	0		0	0	0	0	0	1		0	0		0	0	0
01111	0	0			0	0	0	1		0	0			0		1	0	0		0		
10000	0	0			0	1	0	0		0	0			0			0	0		0		
10001	0	0			0	0	0	0		0	0			0			1	0		0		
10010	0	0			0	0	0	0	0	0	0			0			0	1		0		
10011	0	0			0	0	0	0		0	0			0			0	0	1	1		
10100	0	0			0	0	0	0		0	0			0			0	0	0	1		

00000 00001 00010 00011 00100 00101 00110 01011 01100 01110 10000 00111 01000 01111 01001 01010 01011 01100 01101 01110 01111 10001 10010 10011 10100 00101 00110

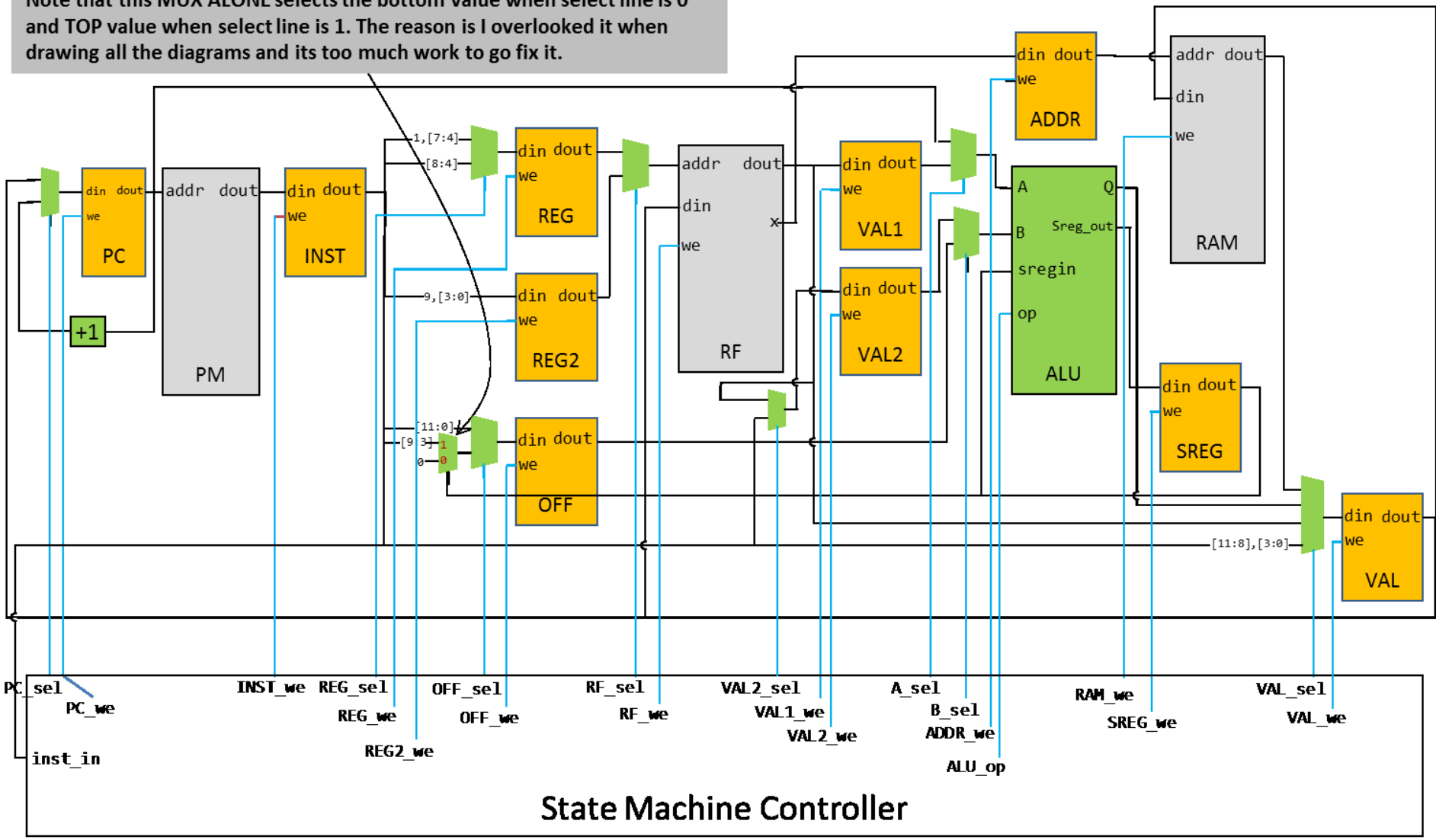


State#	Name	PC_sel	PC_we	INST_we
10011	INST=PM[PC]	0	0	1
10011	PC = PC+1	1	1	0
10100	PC = VAL	0	1	0





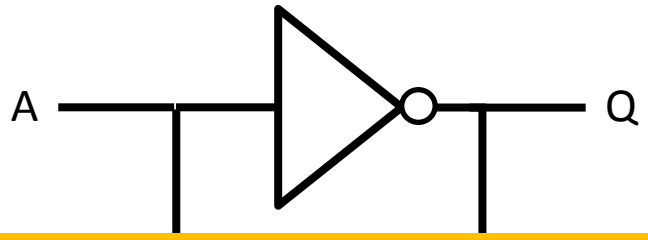
Note that this MUX ALONE selects the bottom value when select line is 0 and TOP value when select line is 1. The reason is I overlooked it when drawing all the diagrams and its too much work to go fix it.



Today

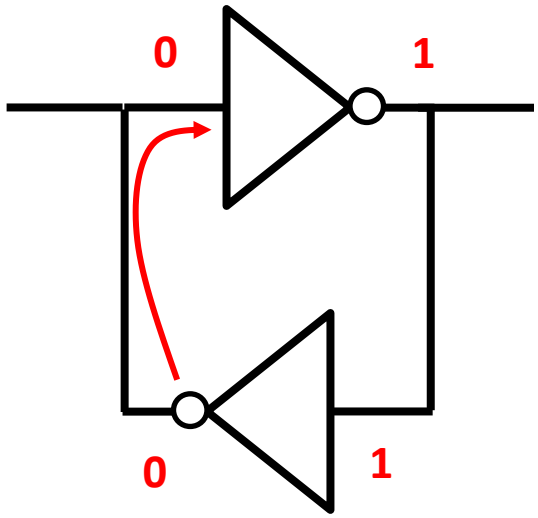
- Today: Sequential elements
  - 1-bit memory cell
  - 8-bit auxilliary register
  - Register files and memories

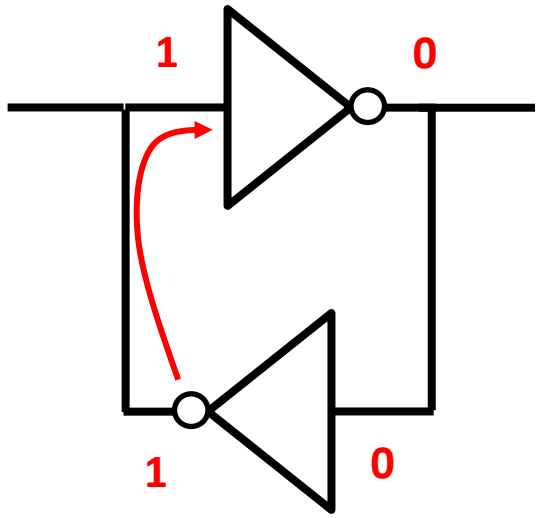
Cross-coupled invertors



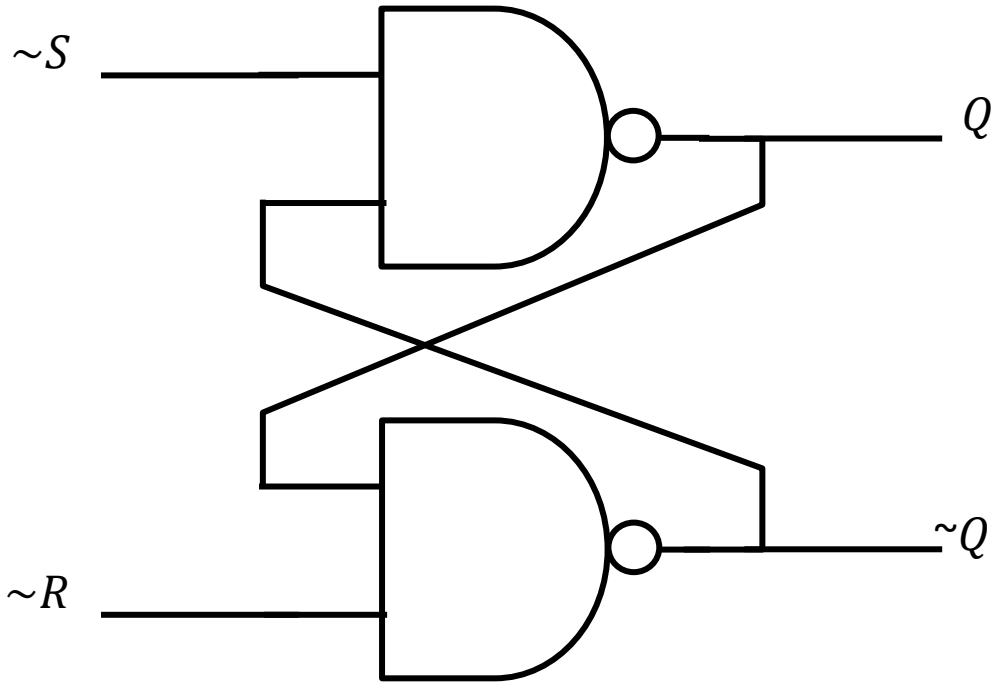
Good news: holds on to value

Bad news: How do we write to it?

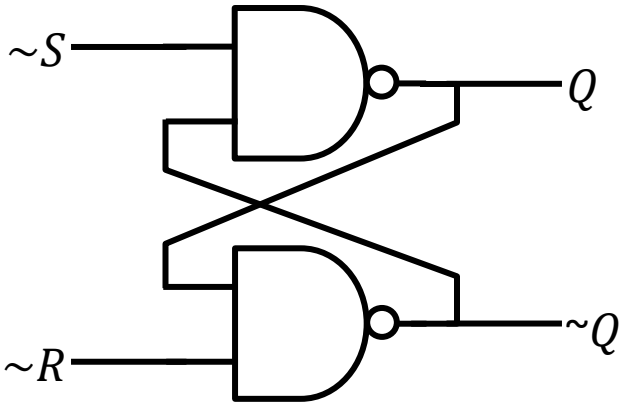




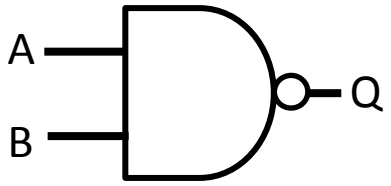
RS Latch



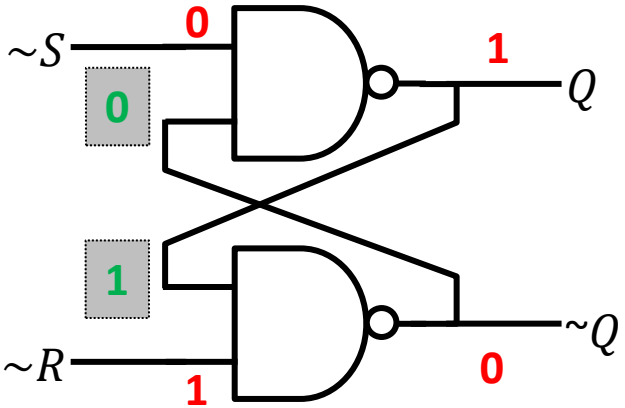
RS Latch



NAND GATE

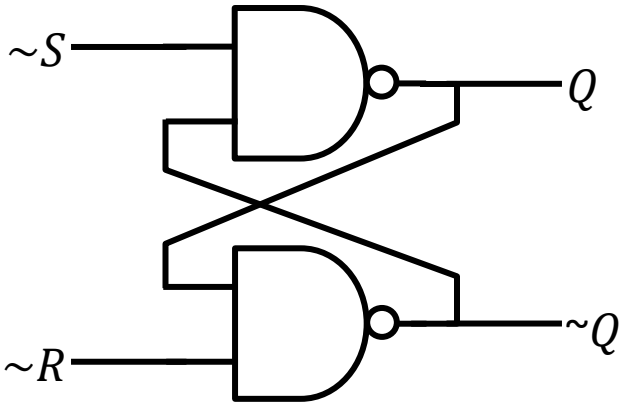


A	B	Q
0	0	**1**
0	1	**1**
1	0	**1**
1	1	0

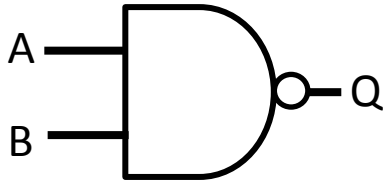


	0 ps	…	10ps	…	20ps	
~S	**0**	**0**	**0**			**0**
~R	**1**	**1**	**1**			**1**
Q	*0 or 1*	*0 or 1*	**1**		**1**	1
~Q	*0 or 1*	*0 or 1*	*0 or 1*		**0**	**0**

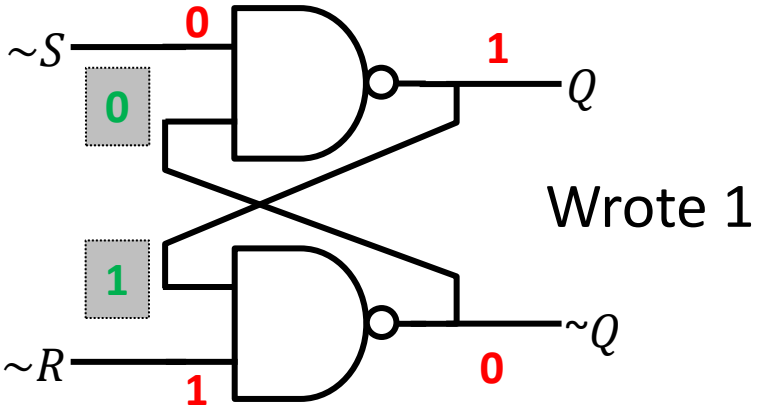
RS Latch



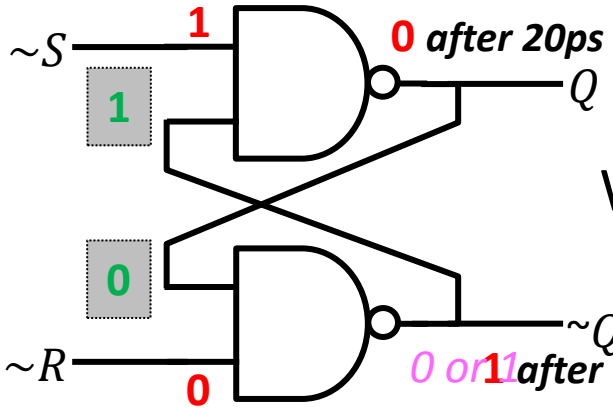
NAND GATE



A	B	Q
0	0	**1**
0	1	**1**
1	0	**1**
1	1	0

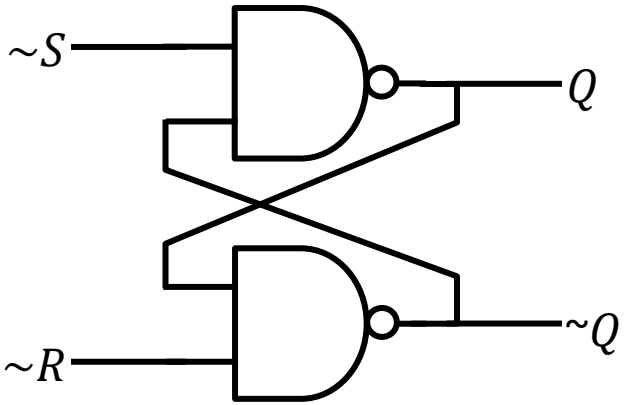


Wrote 1

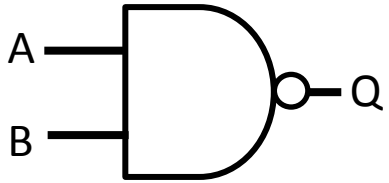


Wrote 0

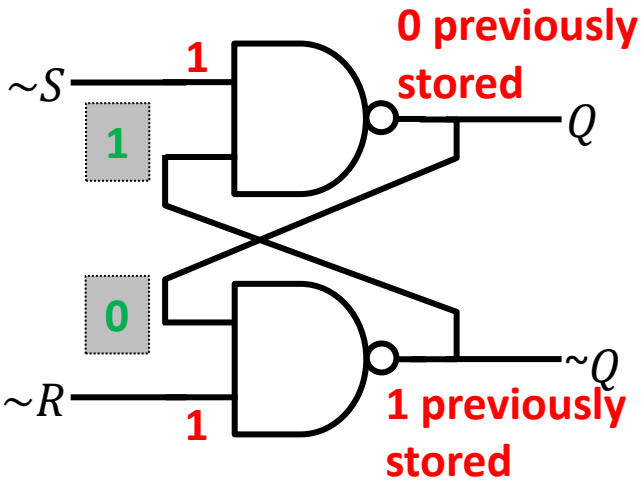
RS Latch



NAND GATE

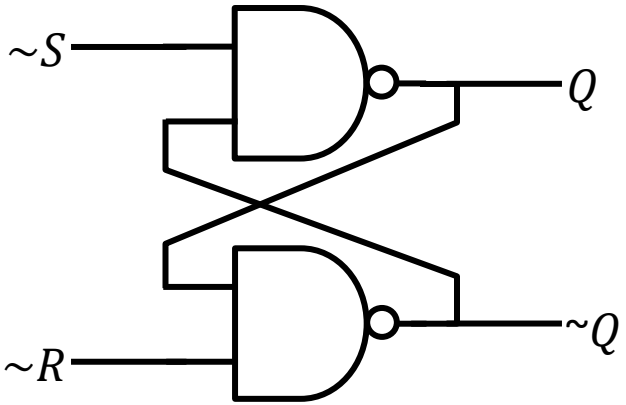


A	B	Q
0	0	**1**
0	1	**1**
1	0	**1**
1	1	0

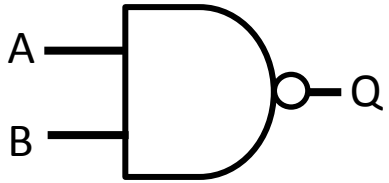


Maintain or store 0 (previous value)

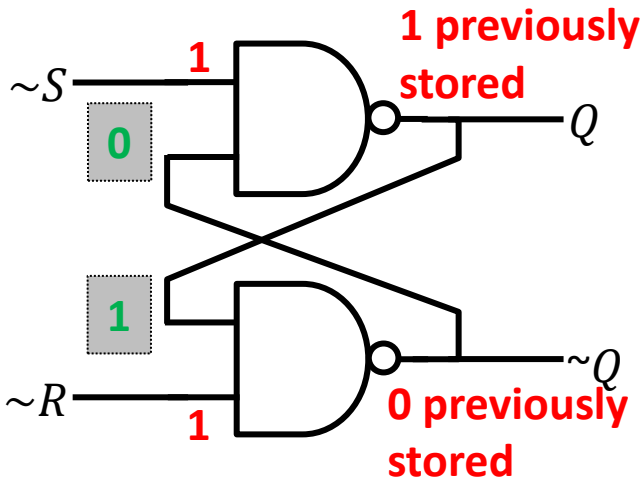
RS Latch



NAND GATE

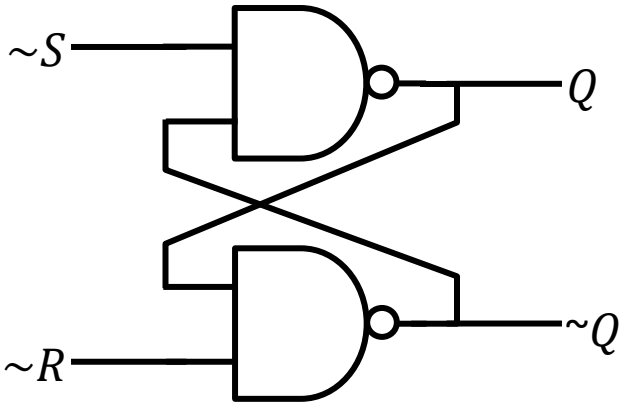


A	B	Q
0	0	**1**
0	1	**1**
1	0	**1**
1	1	0

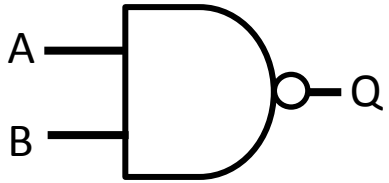


Maintain or store 1 (previous value)

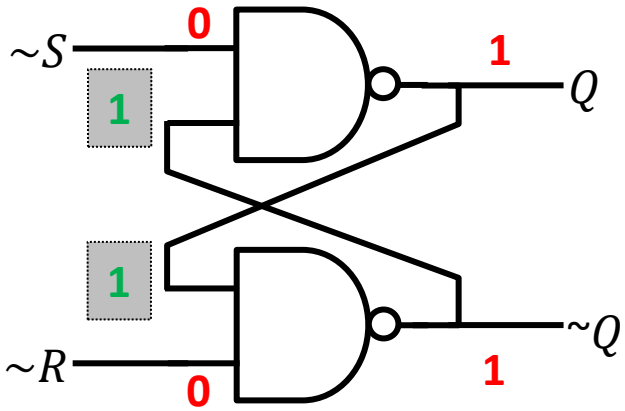
RS Latch



NAND GATE



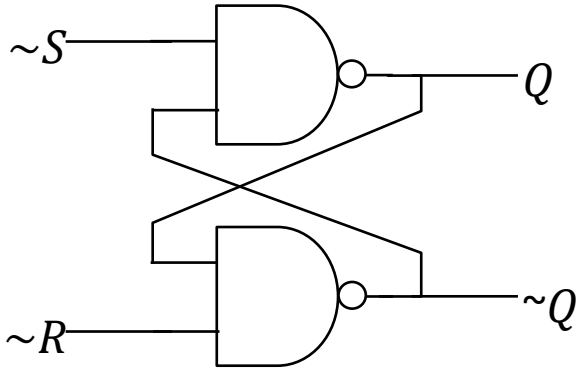
A	B	Q
0	0	**1**
0	1	**1**
1	0	**1**
1	1	0



Q and ~Q are 1!
So would like to "disallow"

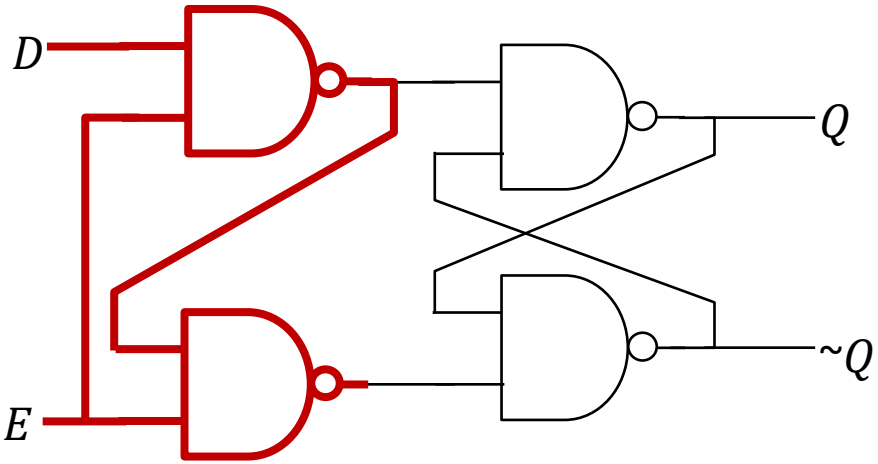
RS Latch

$\sim S$	$\sim R$	Action
0	0	Not allowed or undefined
0	1	Q = 1; store 1
1	0	Q = 0; store 0
1	1	**No change**



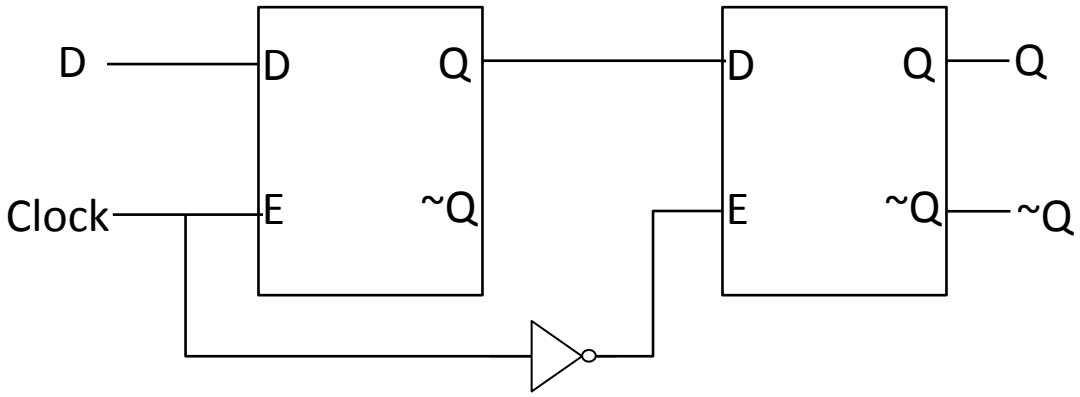
Gated D-latch

E	D	Q	Action
0	X	Q(previous)	**No change**
1	0	0	Reset
1	1	1	Set

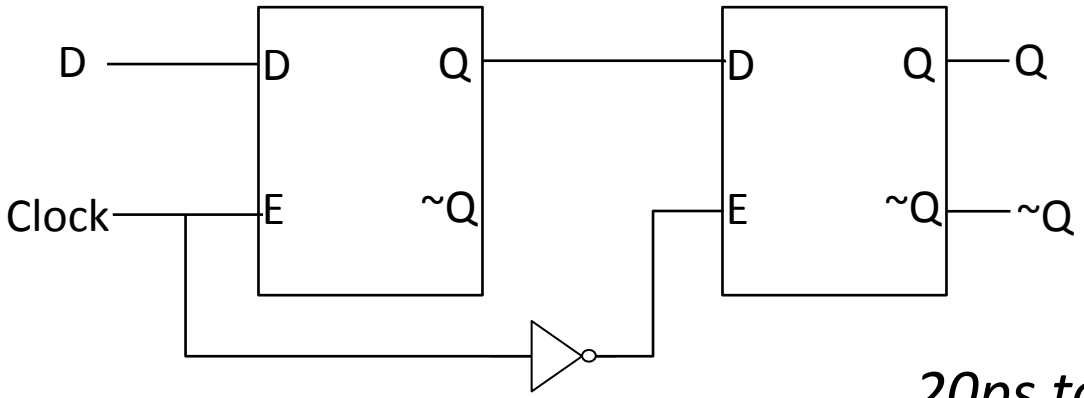


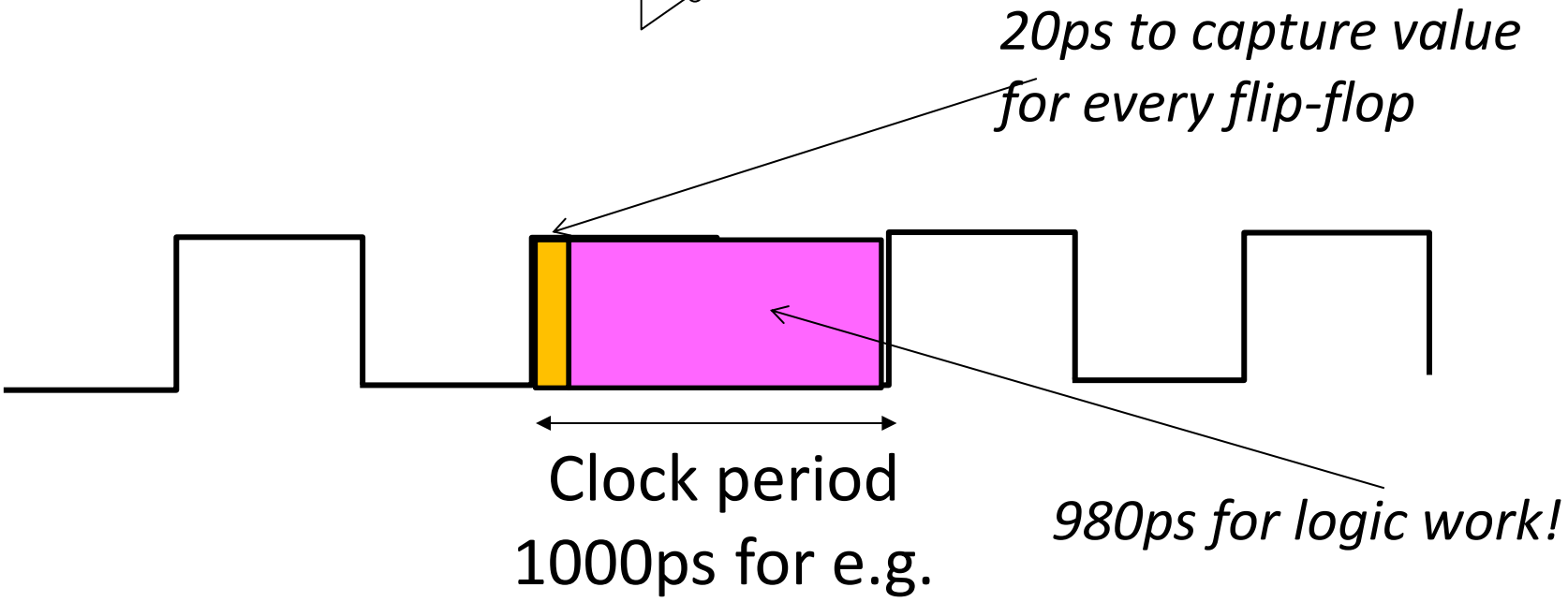
- When E is 0,
  - The two red NAND gates are forced to output 1.
  - Gets us to the 1-1 hold state for the RS Latch.
- When E is 1, D becomes the controlling input.
  - When D is 0 we write 0 into the RS-latch
  - When D is 1 we write 1 into the RS-latch

Master Slave Flip Flop

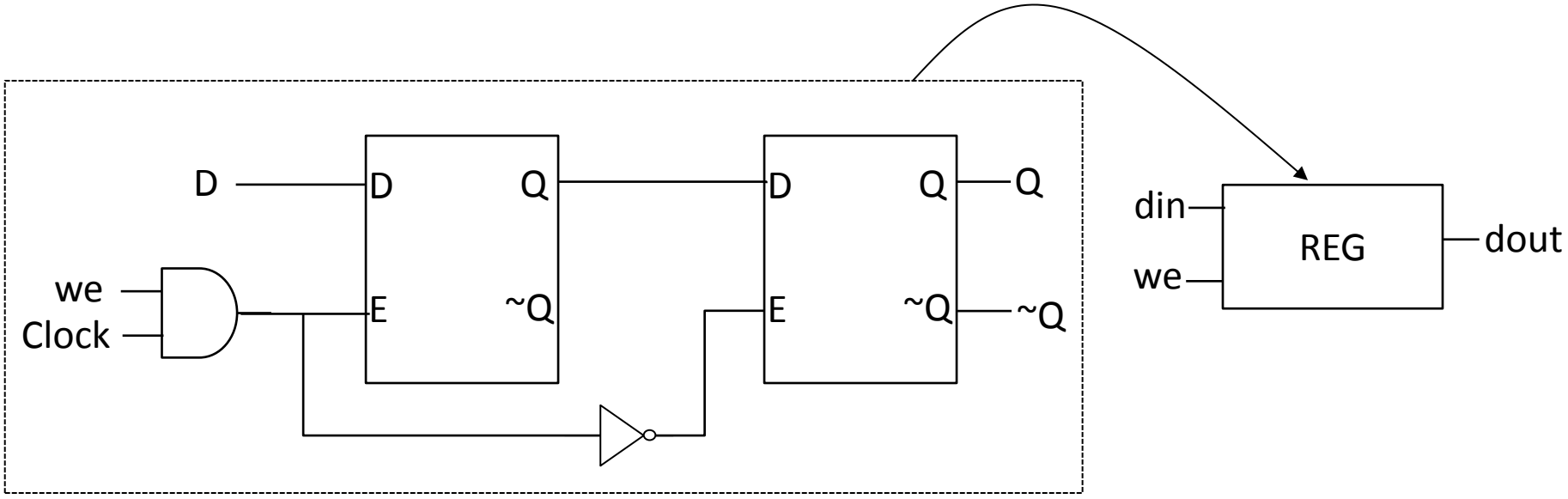


Master Slave Flip Flop



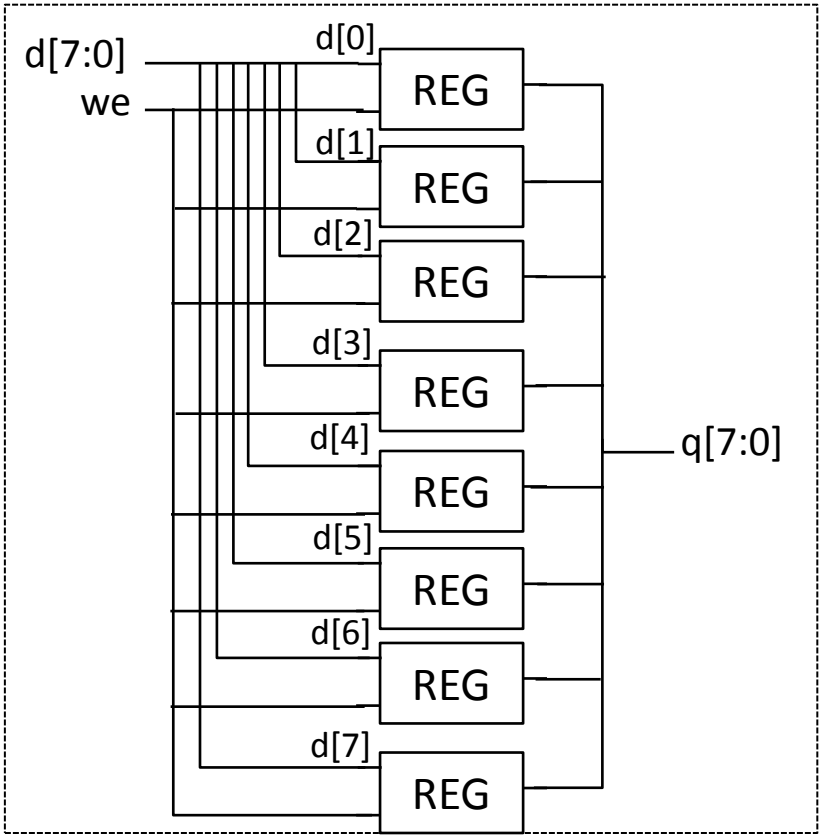


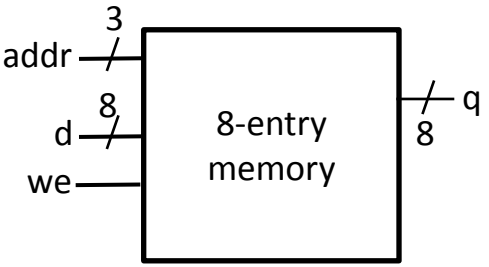
Master-Slave Flip-Flop with WE

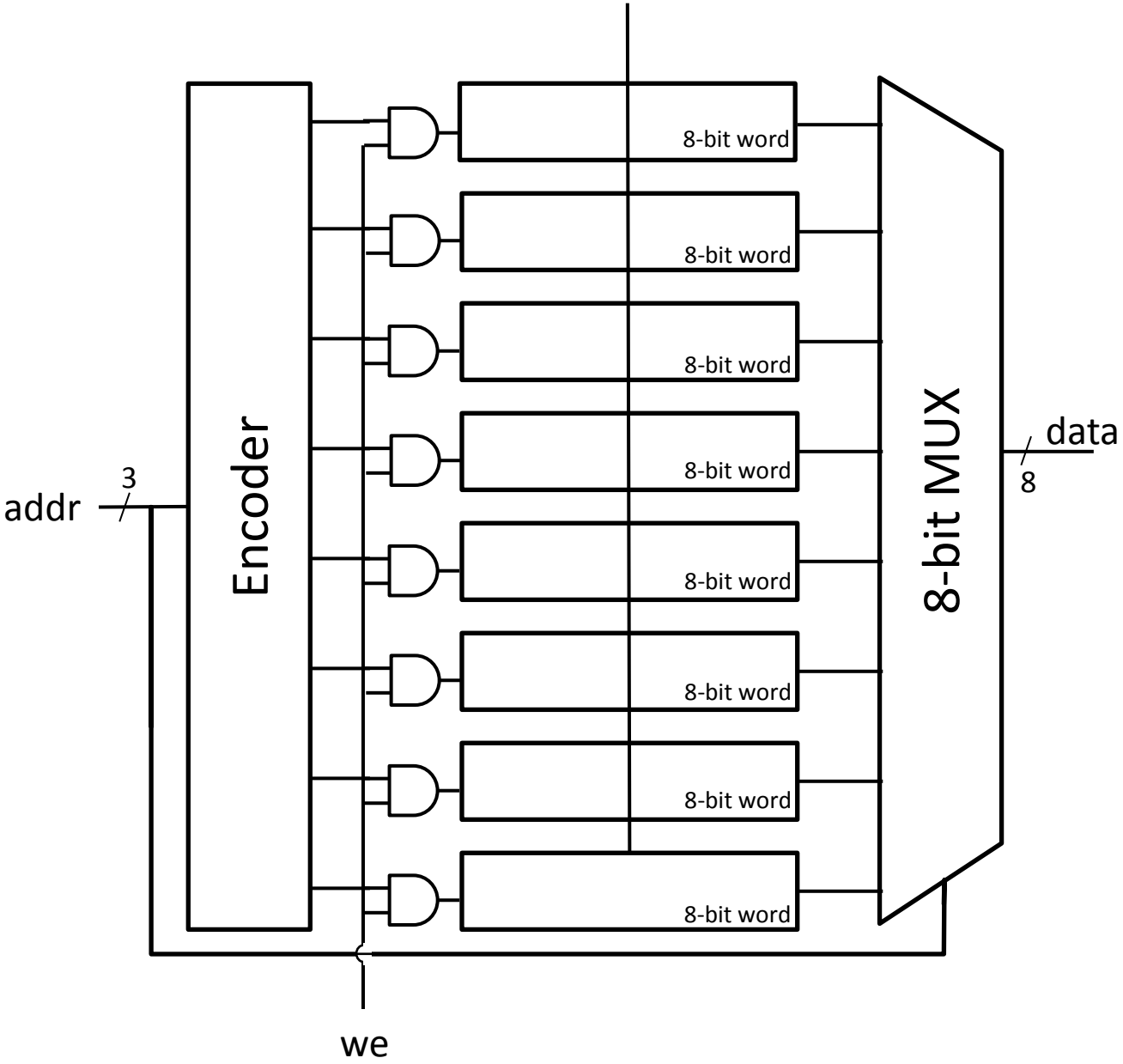


Clock is an implicit
Input hooked up to all REGs

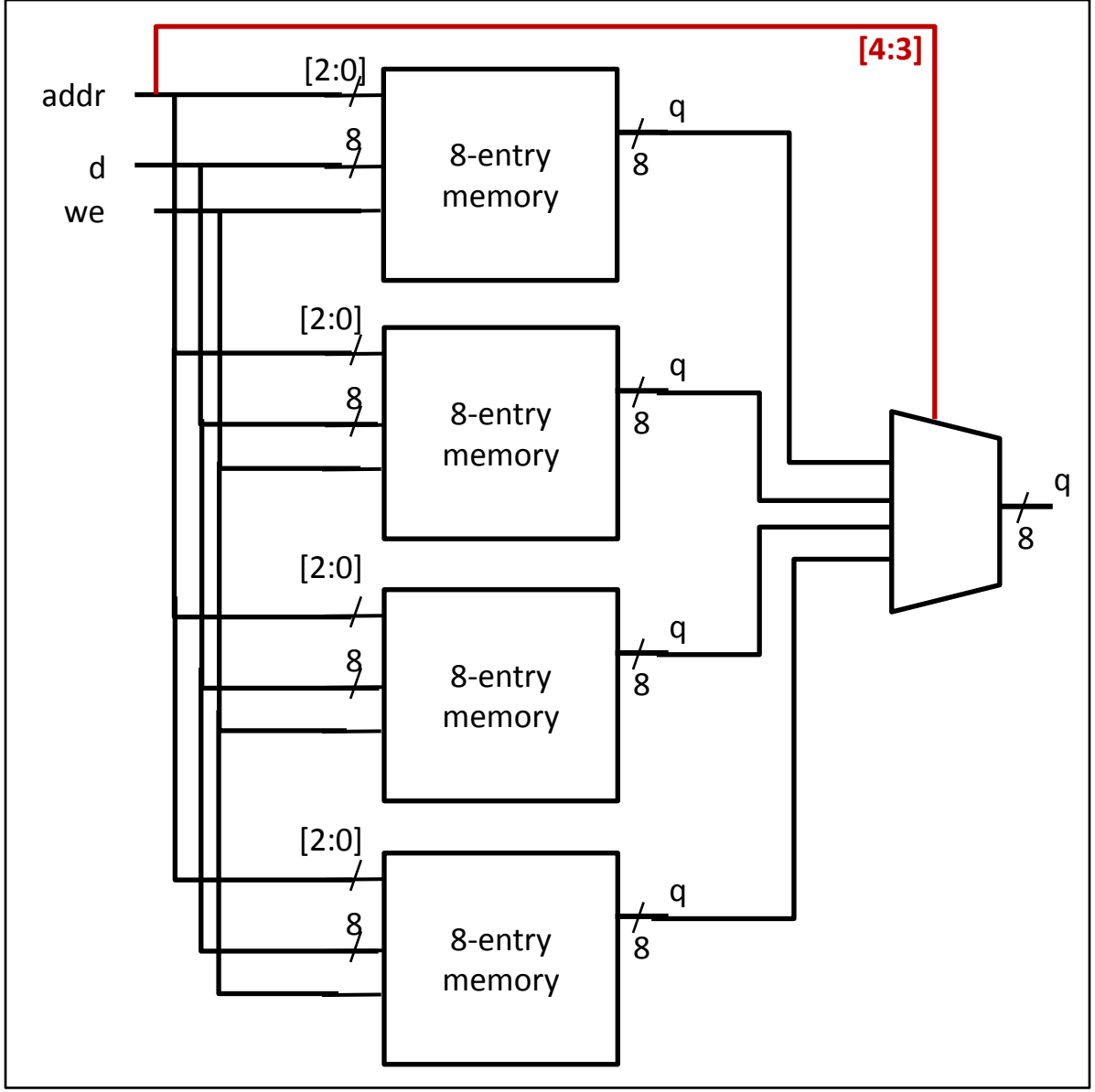
Auxiliary Register

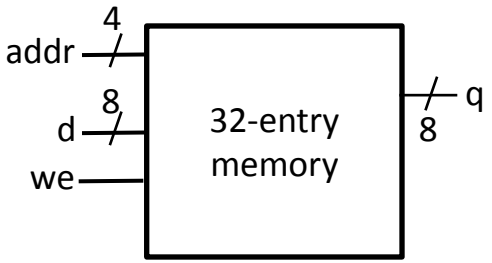




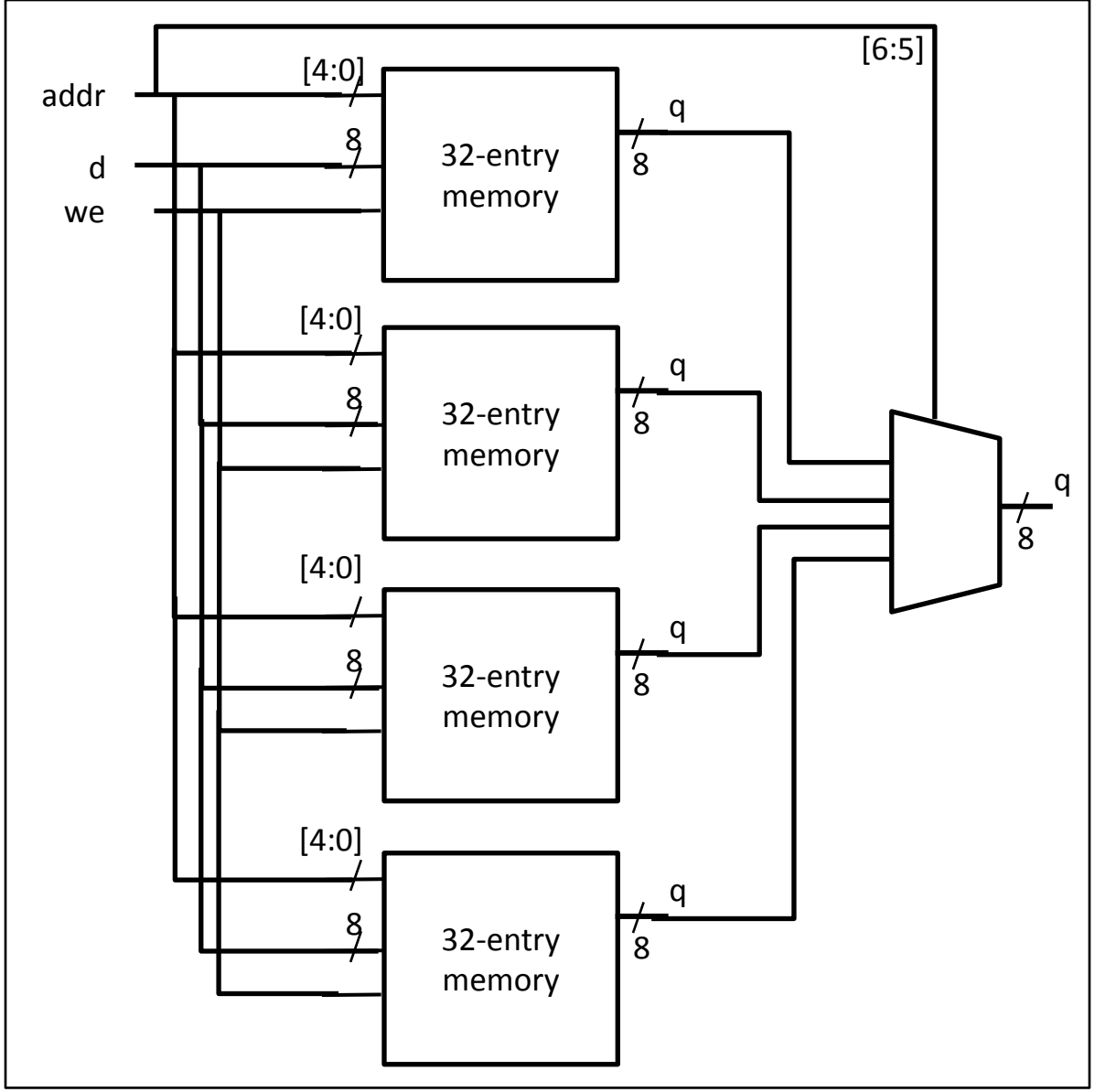


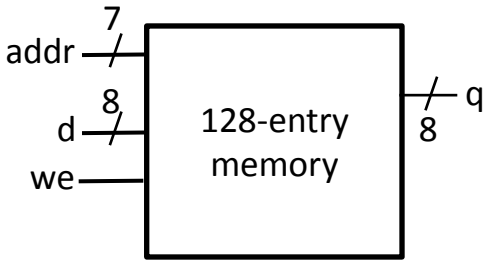
8-entry 8-bit memory





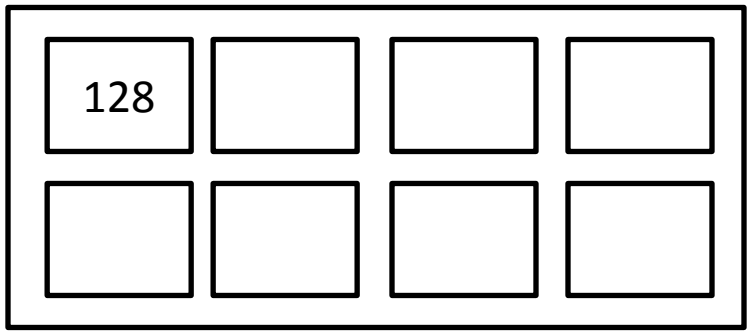
RF (32 entry memory)





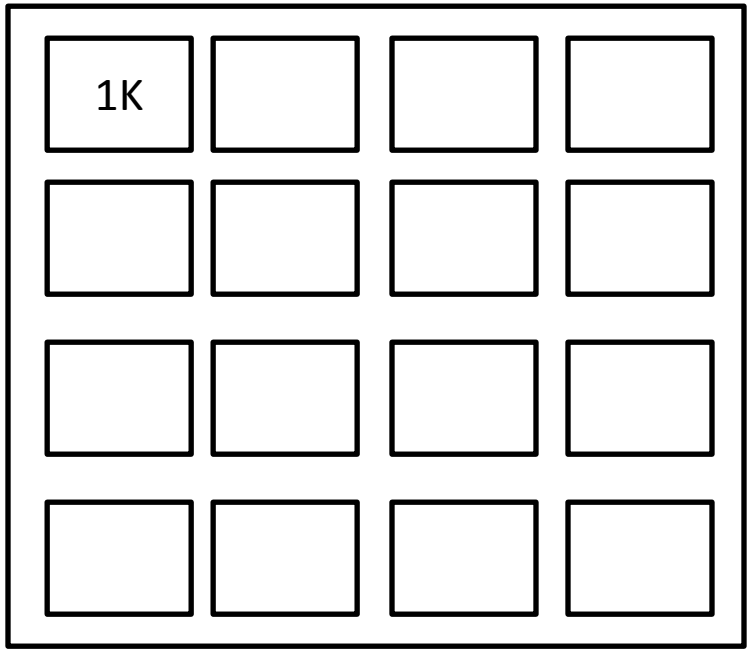
128 entry memory

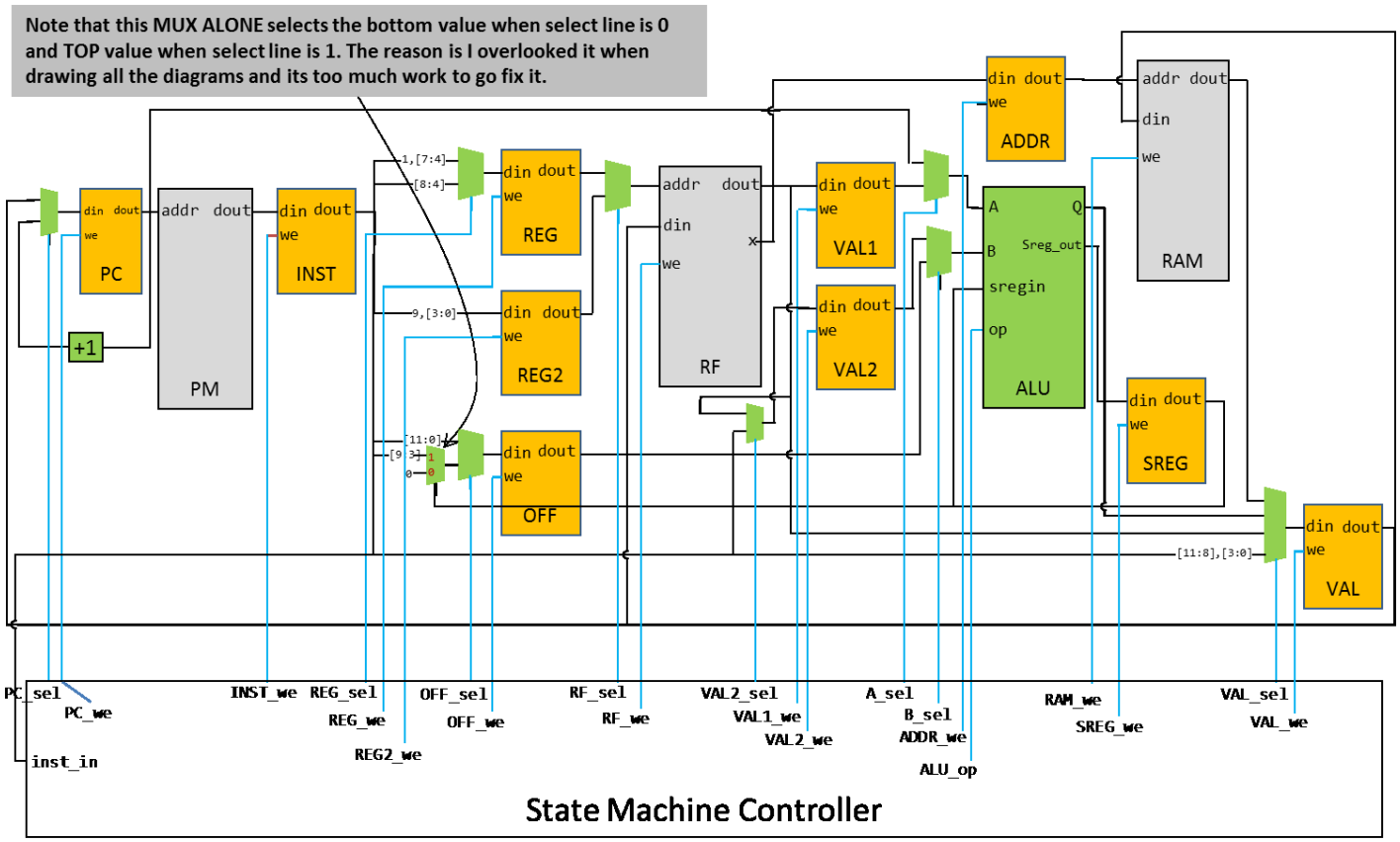
16K Memory



1K memory

16K memory





- Memories (PM, RF, RAM)
- Registers
- MUX (done)
- ALU (done)
- Incrementor (done)

Today

- Today: Sequential elements
  - 1-bit memory cell
  - 8-bit auxiliary register
  - Register files and memories
- Monday
  - Transistors
  - In-class short book survey (come prepared with knives sharpened!)