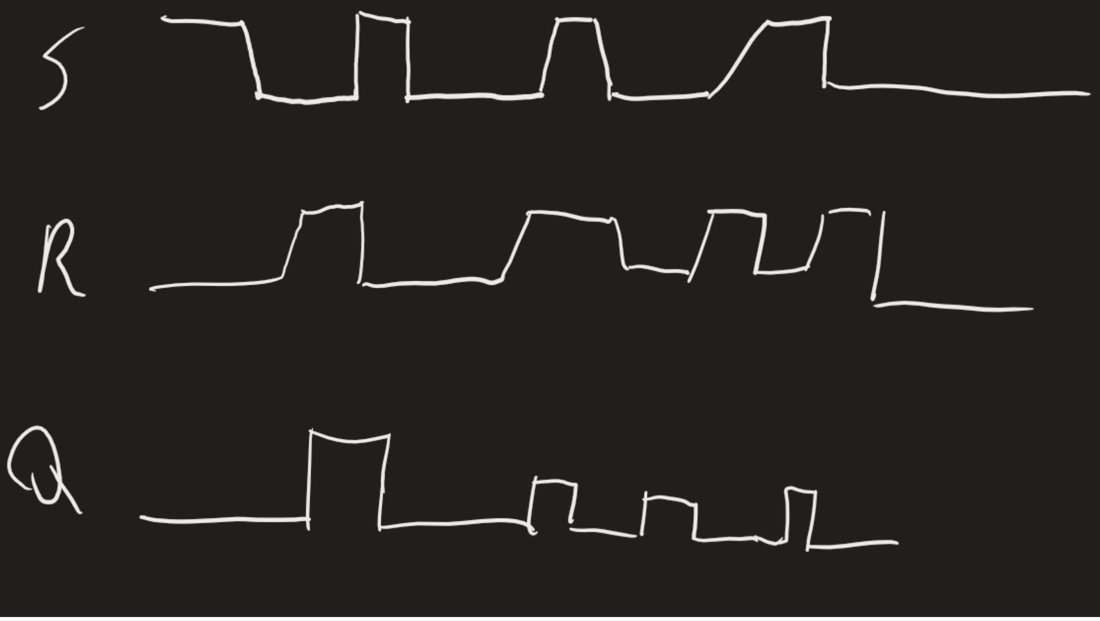
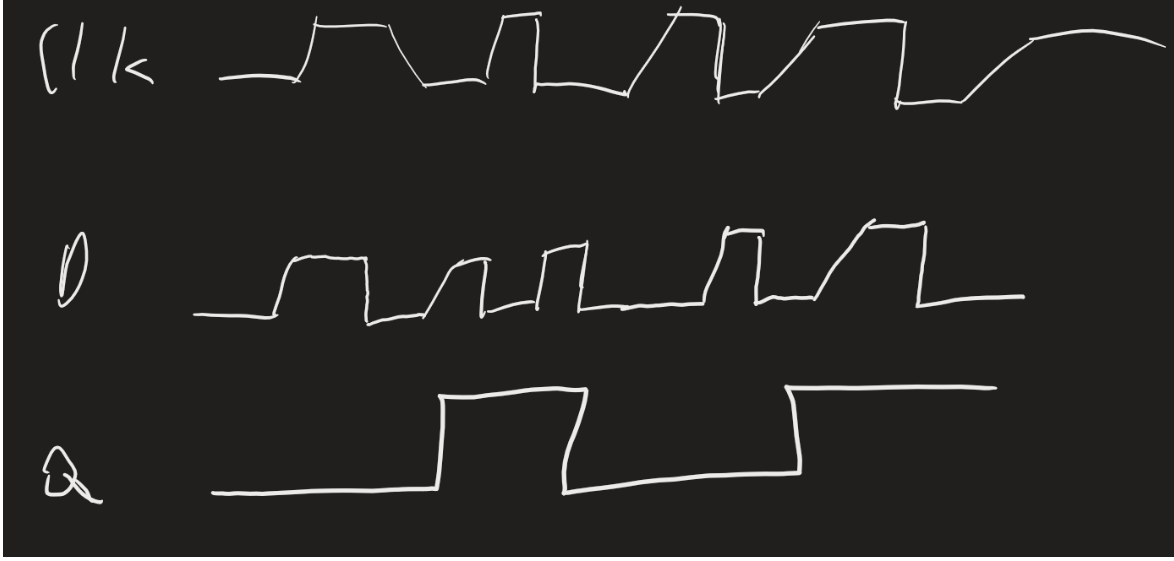
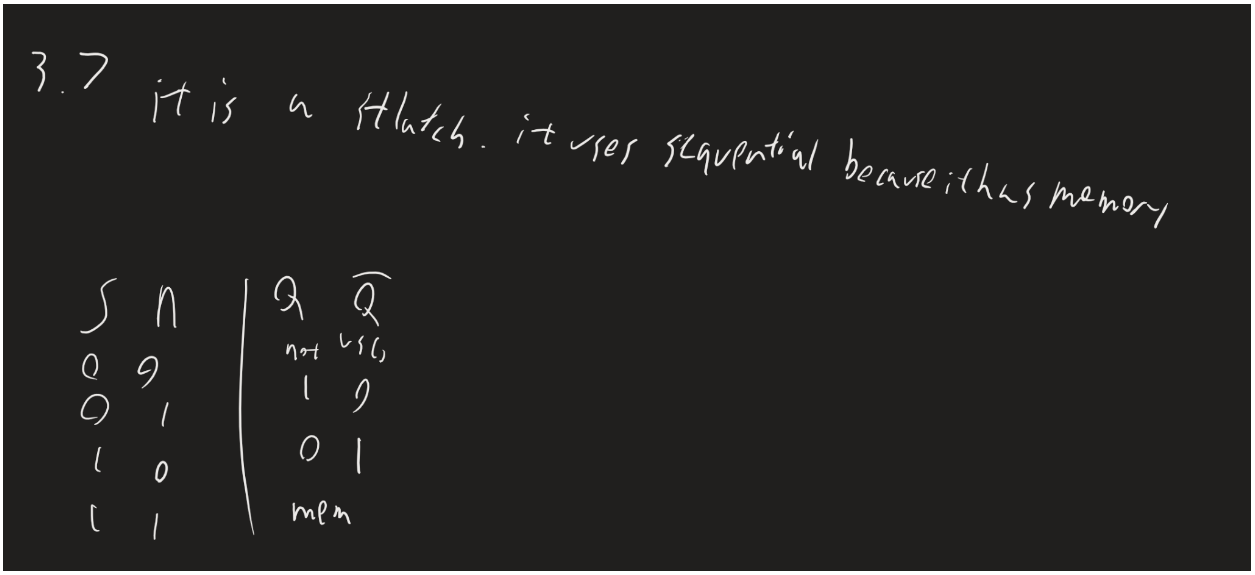
2.2)



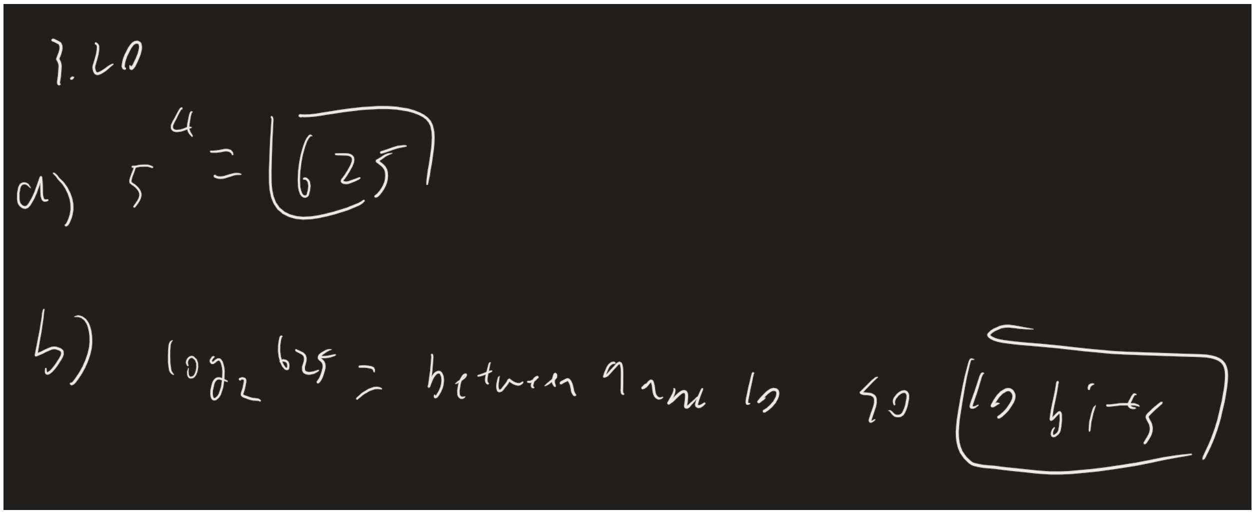
3.4





3.7 it is a SRlatch. it uses sequential because it has memory

S	R	Q	$\overline{Q}$
0	0	not	used
0	1	1	0
1	0	0	1
1	1	mem	

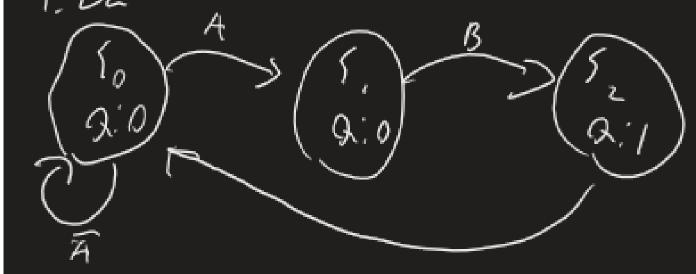


3.20

a) $5^4 = \boxed{625}$

b) $\log_2 625$ = between 9 and 10 so $\boxed{\text{10 bits}}$

7.22



a state machine is a machine that uses inputs to change their outputs.

Cur state	inputs	next state
S_0	A	S_1
S_1	B	S_2
S_2	X	S_0
S_1	$\bar{B}$	S_0
S_0	$\bar{A}$	S_0

output table!

Cur state	next state
S_0	S_1
S_1	S_2
S_2	S_0
S_1	S_0
S_0	S_0

$$S_1(S_0 A) + S_2(S_1 B) + S_0(S_2 + S_1 \bar{B} + S_0 \bar{A})$$

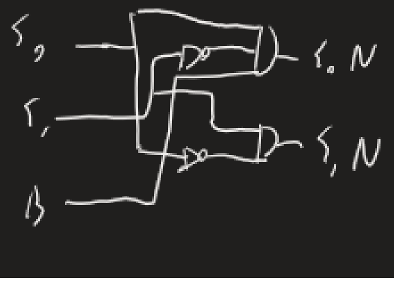
S_1	S_0	Input	S_1	S_0
0	0	A	0	0
0	1	B	0	1
1	0	X	1	0
0	0	$\bar{A}$	0	0
0	0	$\bar{B}$	0	0

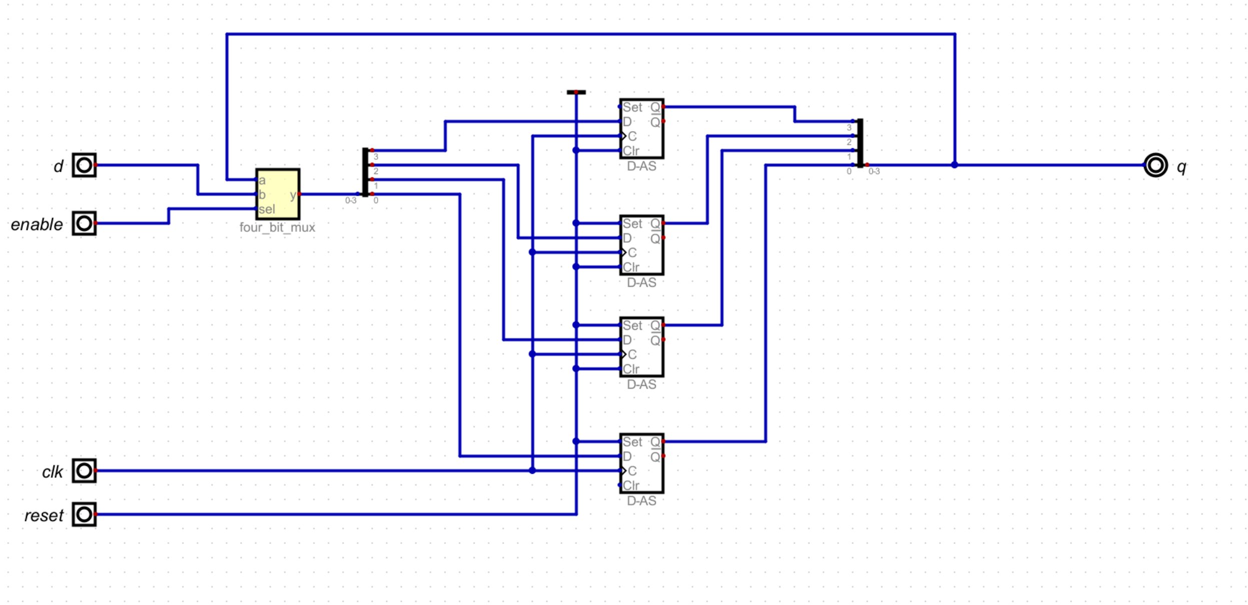
$Q = S_1$

$S_0 = \bar{S_1} S_0 B$

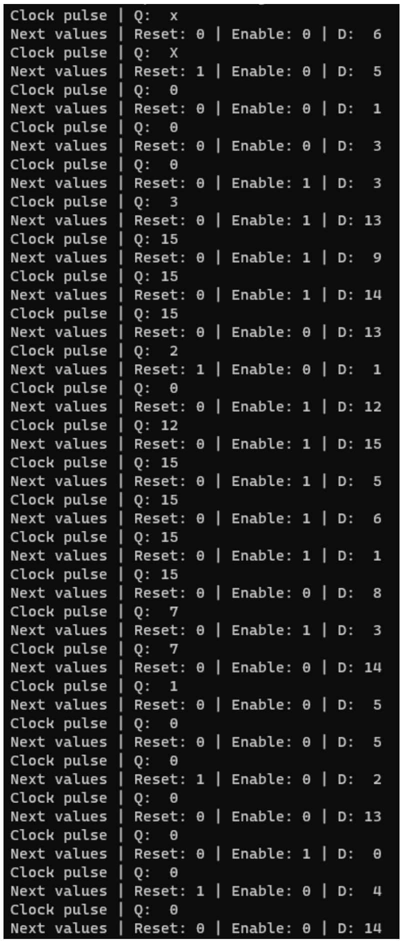
$S_1 = S_1 \bar{S_0}$

S_1 \ S_0	0	1
0	0	0
1	1	-

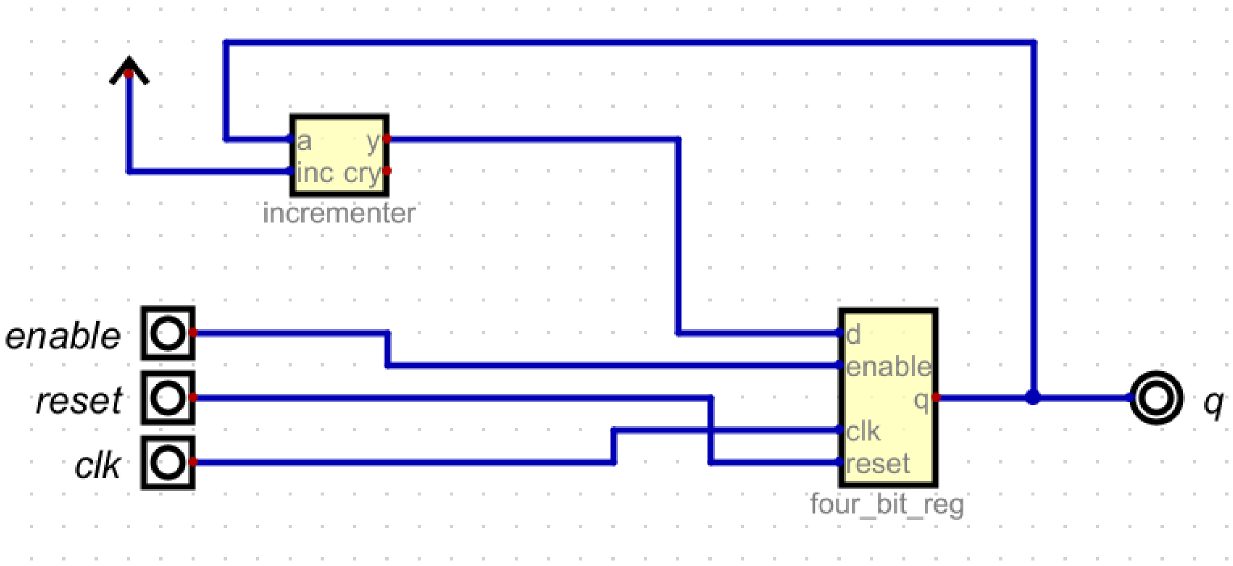




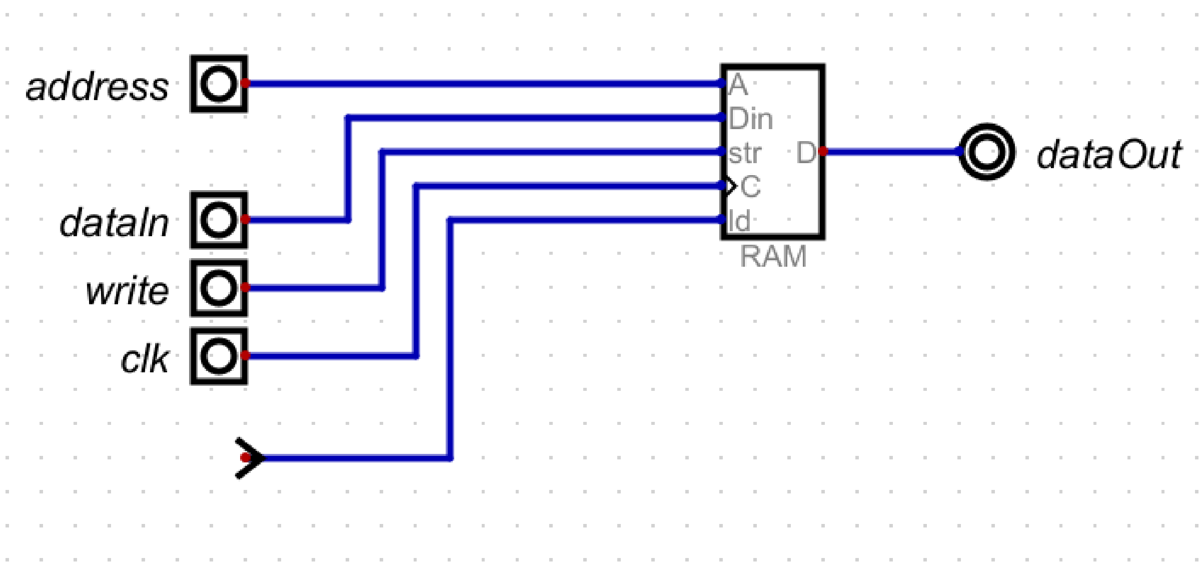
This is a 4 bit register that uses d-flip-flip switches that are asynchronous. The output is a 4 bit q.



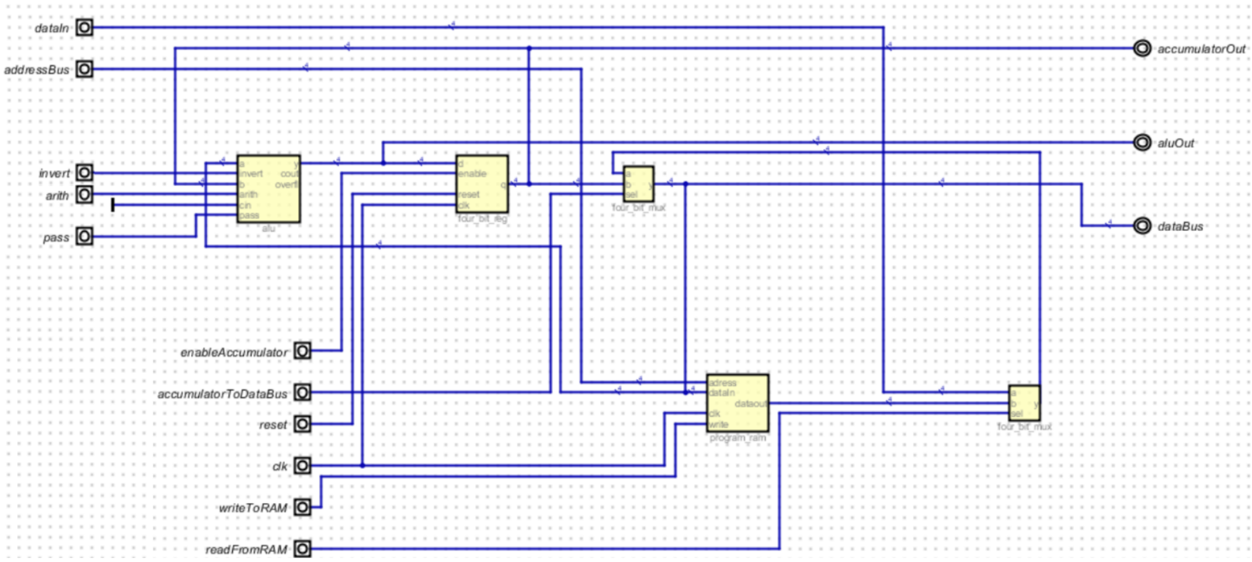
```
Clock pulse | Q:  x
Next values | Reset: 0 | Enable: 0 | D:  6
Clock pulse | Q:  X
Next values | Reset: 1 | Enable: 0 | D:  5
Clock pulse | Q:  0
Next values | Reset: 0 | Enable: 0 | D:  1
Clock pulse | Q:  0
Next values | Reset: 0 | Enable: 0 | D:  3
Clock pulse | Q:  0
Next values | Reset: 0 | Enable: 1 | D:  3
Clock pulse | Q:  3
Next values | Reset: 0 | Enable: 1 | D: 13
Clock pulse | Q: 15
Next values | Reset: 0 | Enable: 1 | D:  9
Clock pulse | Q: 15
Next values | Reset: 0 | Enable: 1 | D: 14
Clock pulse | Q: 15
Next values | Reset: 0 | Enable: 0 | D: 13
Clock pulse | Q:  2
Next values | Reset: 1 | Enable: 0 | D:  1
Clock pulse | Q:  0
Next values | Reset: 0 | Enable: 1 | D: 12
Clock pulse | Q: 12
Next values | Reset: 0 | Enable: 1 | D: 15
Clock pulse | Q: 15
Next values | Reset: 0 | Enable: 1 | D:  5
Clock pulse | Q: 15
Next values | Reset: 0 | Enable: 1 | D:  6
Clock pulse | Q: 15
Next values | Reset: 0 | Enable: 1 | D:  1
Clock pulse | Q: 15
Next values | Reset: 0 | Enable: 0 | D:  8
Clock pulse | Q:  7
Next values | Reset: 0 | Enable: 1 | D:  3
Clock pulse | Q:  7
Next values | Reset: 0 | Enable: 0 | D: 14
Clock pulse | Q:  1
Next values | Reset: 0 | Enable: 0 | D:  5
Clock pulse | Q:  0
Next values | Reset: 0 | Enable: 0 | D:  5
Clock pulse | Q:  0
Next values | Reset: 1 | Enable: 0 | D:  2
Clock pulse | Q:  0
Next values | Reset: 0 | Enable: 0 | D: 13
Clock pulse | Q:  0
Next values | Reset: 0 | Enable: 1 | D:  0
Clock pulse | Q:  0
Next values | Reset: 1 | Enable: 0 | D:  4
Clock pulse | Q:  0
Next values | Reset: 0 | Enable: 0 | D: 14
```



This is a 4 bit reg that increments from its output. The clock controls when the data is transferred. The reset clears the contents.



This is an example of how ram works. The address of the data, the data itself, the clock to control the speed, and write to change the data are all inputs for the ram circuit.



DataIn is the raw input data that may be redirected back to the program RAM.
AddressBus is the selector that chooses which four bits of the program_ram will be accessed.
Invert is the operation that flips the bits of the DataIn using the alu and sends the result to the ALUout.
Arith is the operation that performs arithmetic or bitwise functions on the DataIn using the alu and sends the result to the ALUout.
Pass is the operation that passes the DataIn and AddressBus values to the ALUout and DataBus outputs without any changes.
EnableAccumulator is the signal that triggers the four_bit_reg to store the DataIn value in its register.
Reset is the signal that resets the four_bit_reg value to 0.
WriteToRam is the process that writes data from the CPU to a specific memory address in the program_ram using the AddressBus and DataIn values.
ReadFromRam is the process that reads data from a specific memory address in the program_ram and sends it to the DataBus output using the AddressBus value.