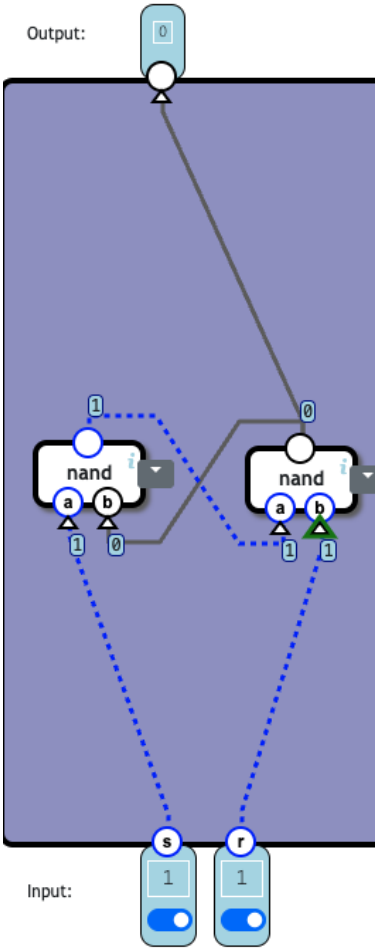
1. SR Latch

Input		Output
s	**r**	
1	0	1
0	1	0
1	1	Previous output
0	0	*Not used*

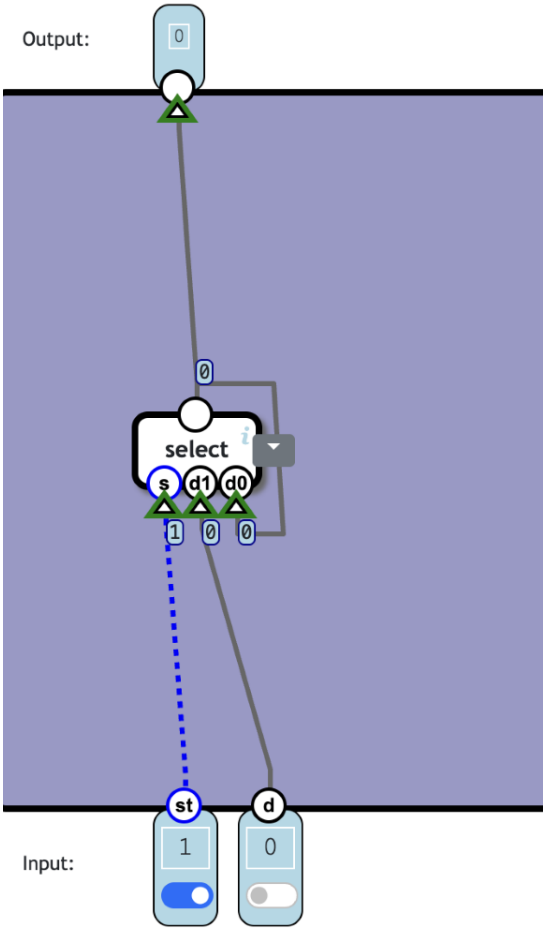
Undefined initial output
0, 1 --> 1, 1 outputs 0
1, 0 --> 1, 1 outputs 1

Route s and r to the input of their own NAND gate
The remaining input of each gate is the output of the other.



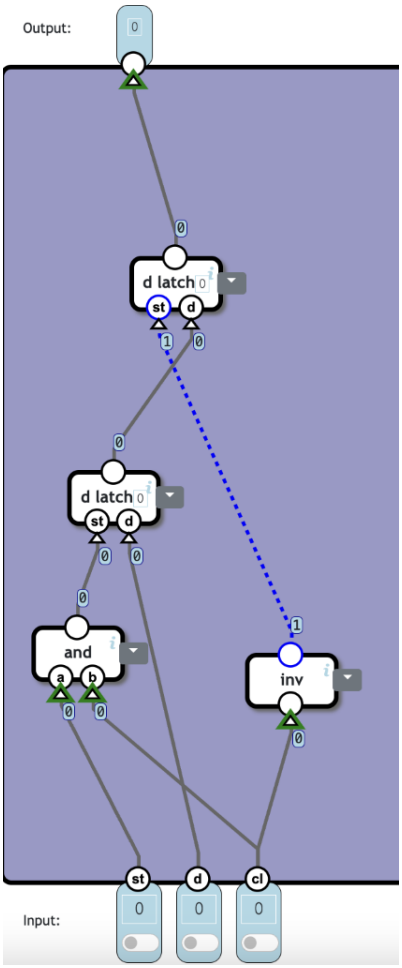
2. D Latch (Data Latch)

"The SR latch is somewhat impractical as a building block for memory because of the separate pins for storing 1 or 0 and because of the “forbidden” (0, 0) input state. The Data Latch extends it with a more practical input: One input for data and an input to indicate if the data should be stored."

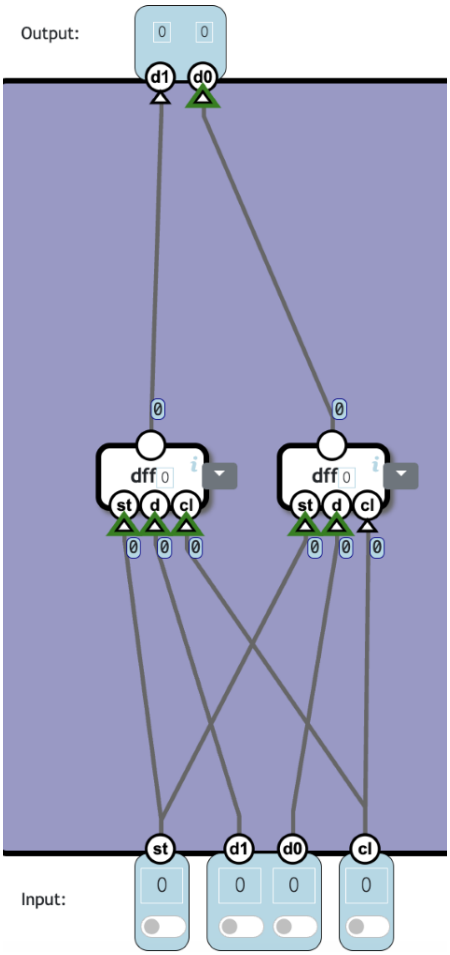


3. Data Flip-Flop

Writes to first latch when cl = 1 and st = 1
Writes to second latch from first latch when cl = 0

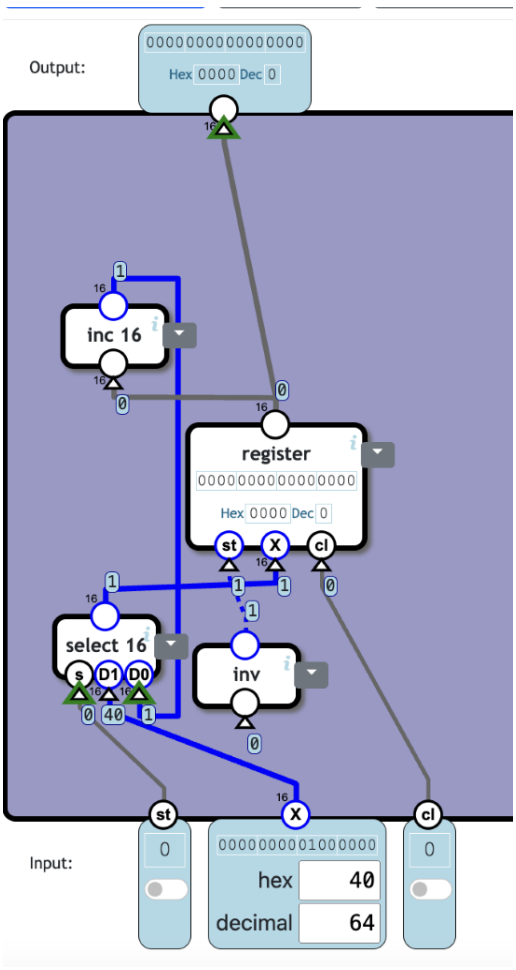

Output:
0
d latch
st
d
1
0
d latch
st
d
and
a
b
inv
st
d
cl
Input:
0
0
0

4. Register (2 bit flip-flop)



5. Counter

Increments the time every clock cycle
Outputs X as the new value if st is 1 for a clock cycle

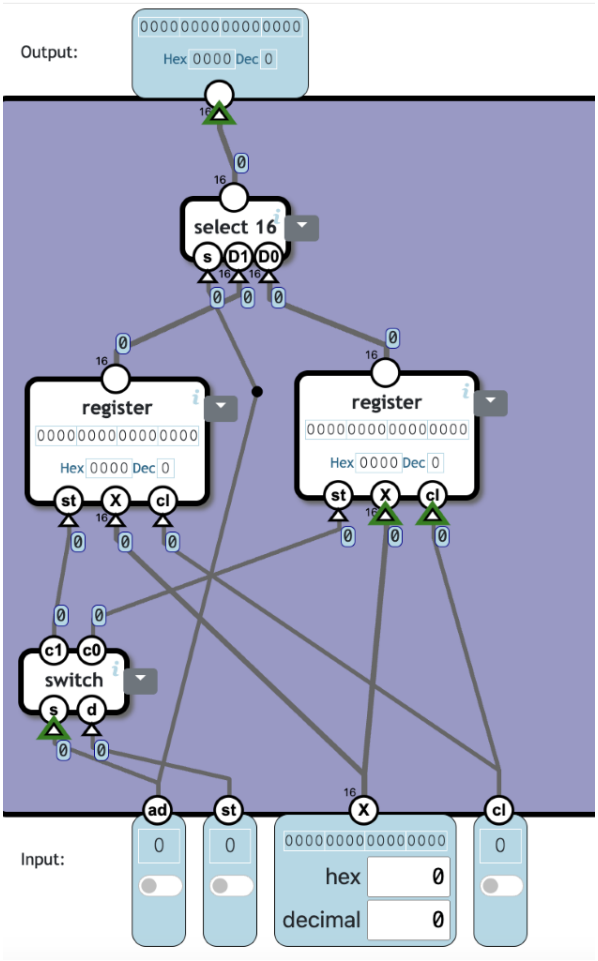


6. RAM

Introduces memory addresses so that individual 16-bit (or other size, by using different amounts of flip-flops) entities can be accessed and modified.

Setting (modifying) and outputting (accessing) is dependent on the state of the add bit

This can be constructed recursively to create any amount of addressable memory (2 bits of ram is 741 NAND gates)



Extra: RAM vs Registers

S.NO.	Register	Memory
1.	Registers hold the operands or instruction that CPU is currently processing.	Memory holds the instructions and the data that the currently executing program in CPU requires.
2.	Register holds the small amount of data around 32-bits to 64-bits.	Memory of the computer can range from some GB to TB.
3.	CPU can operate on register contents at the rate of more than one operation in one clock cycle.	CPU accesses memory at the slower rate than register.
4.	Types are Accumulator register, Program counter, Instruction register, Address register, etc.	Type of memory are RAM, etc.
5.	Registers can be control i.e. you can store and retrieve information from them.	Memory is almost not controllable.
6.	Registers are faster than memory.	RAM is much slower than registers.

Accumulator register or program counter were made when making counter, instruction register and address register are used in the future processor section

Extra: RAM vs ROM

Difference	Random Access Memory (RAM)	Read Only Memory (ROM)
Data-Retention	RAM is a volatile memory that could store the data as long as the power is supplied.	ROM is a non-volatile memory that the could retain the data even when the power is turned off.
Read/Write	Read and write operations are supported.	Only read operations are supported.
Use	Used to store the data that has to be currently processed by CPU temporarily.	It is typically used to store firmware or microcode, which is used to initialize and control hardware components of the computer.
Speed	It is a high-speed memory.	It is much slower than the RAM.
CPU Interaction	CPU can easily access data stored in RAM.	CPU cannot easily access data stored in ROM.
Size and Capacity	Large size with higher capacity, concerning ROM.	Small size with less capacity, concerning RAM.
Used as/in	CPU Cache, Primary memory.	Firmware, Micro-controllers.
Accessibility	The data stored is easily accessible.	The data stored is not as easily accessible as in the concerning RAM.
Cost	RAM is more costlier than ROM.	ROM is cheaper than RAM.
Chip Size	A RAM chip can store only a few gigabytes (GB) of data.	A ROM chip can store multiple megabytes (MB) of data.
Function	Used for the temporary storage of data currently being processed by the CPU.	Used to store firmware, BIOS, and other data that needs to be retained.

https://www.geeksforgeeks.org/difference-between-register-and-memory/
https://www.geeksforgeeks.org/difference-between-ram-and-rom/