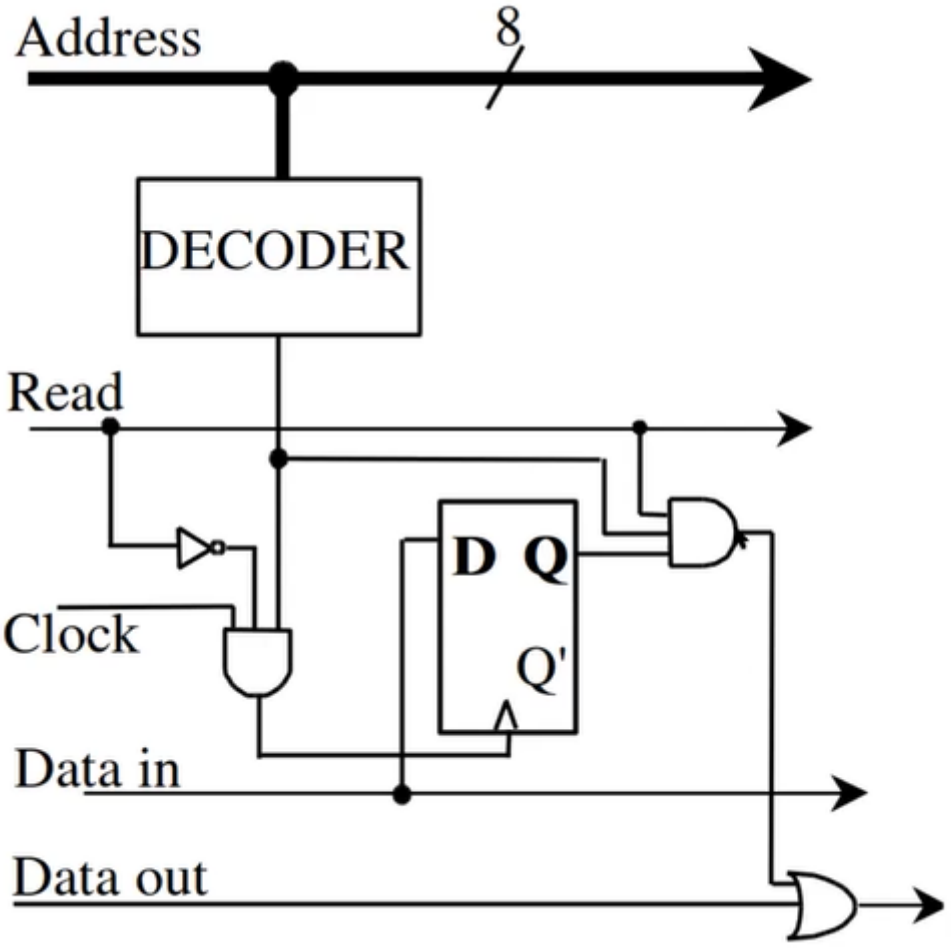
L17 - Random Access Memory and the Fetch Cycle

A single D-type flip flop is a one-bit memory.
To use it as such we need to give it a unique address.
An address is a binary number.
A binary number can be uniquely identified by a decoder.

One Bit Static RAM

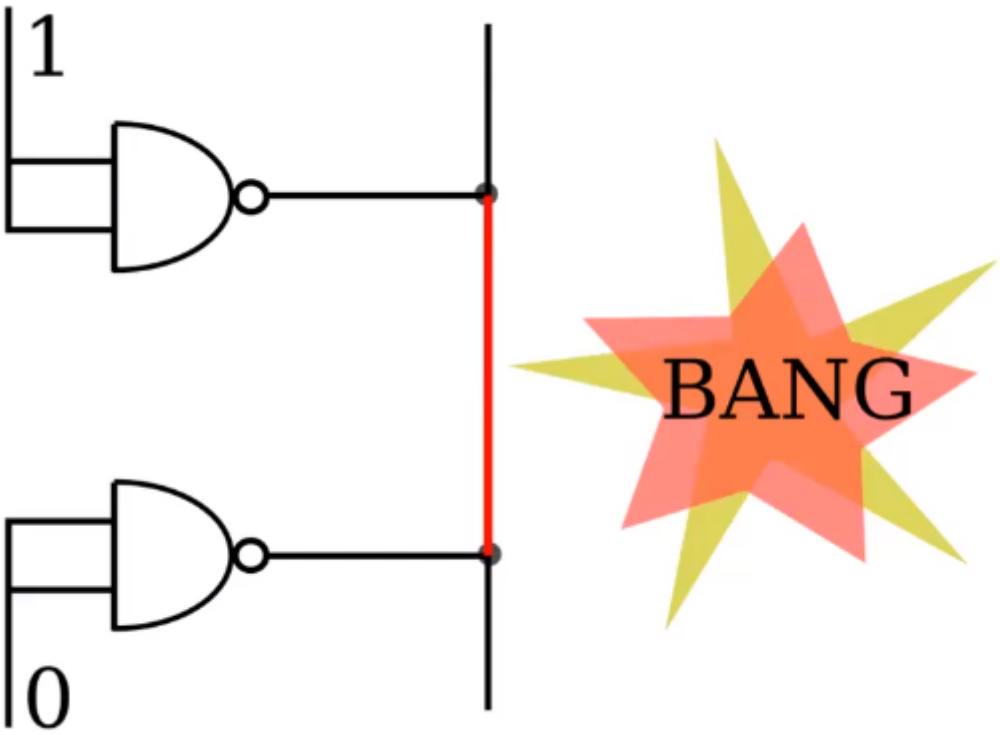


One Bit Static RAM is an assymetric circuit:

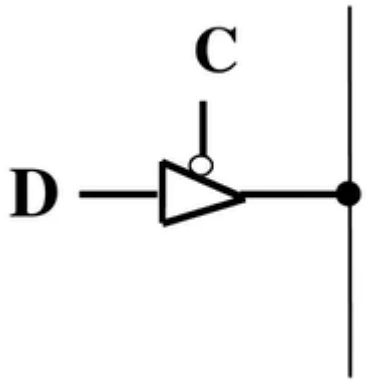
1. For reading, combinatorial circuit
2. For writing, Sequential circuit, the address and data must be present and correct when the clock pulse sets the flip flop
   RAM circuits conforming to this pattern = static RAMs.
   - Only used in special applications because of their large size.

Buses are data highways which are common to many circuits:
- Address bus = The address lines that go to many 1BSRAM circuits
- Data Buses = The data-in and data-out lines from several 1BSRAM circuits are grouped together in bytes
- Control bus = The control lines (read and clock)

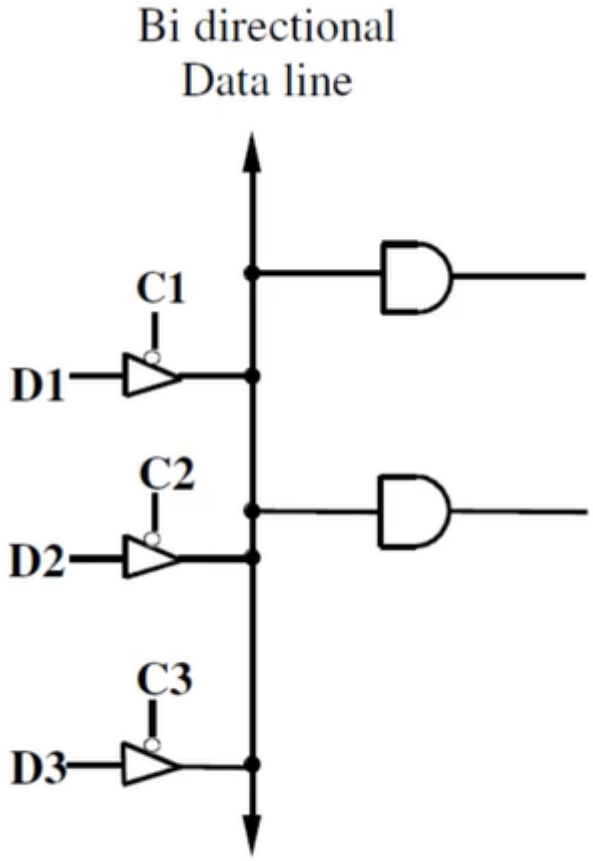
Bi-directional Data Buses:
- In simple processors the data-in and data-out lines are never used at the same time
- In these cases it is convenient to use just one bus as this reduces the size and complexity of the memory circuit
- However, to make the data bi-directional we need to feed it from more than one place and we cannot do this with ordinary AND or NAND gates



There is a special gate called a Tri-State Buffer which can overcome this problem:
- If the control line, C, is set to 0 the output follows the input exactly
- If C = 1, the O is neither 0 nor 1, but is effectively disconnected from the data line

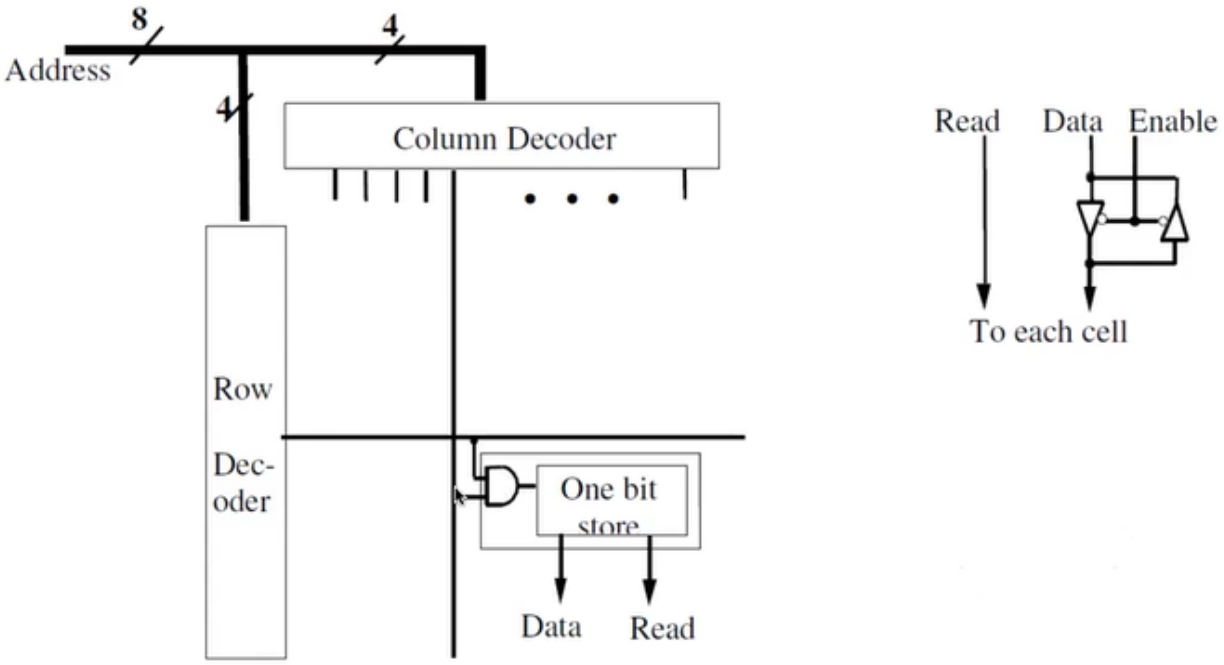


Many different data sources can be fed to a single line providing only one of them has C = 0. Another job for the very versatile demultiplexer!



Physical Layout of RAM

Since RAM is made using a silicon wafer, it is normally organised in 2D with row and column decoders.



Each one bit memory cell is only enabled when both its row and column lines are 1. In the case of a 256 bit RAM each decoder transforms a 4-bit binary number into a sixteen bit unary number.

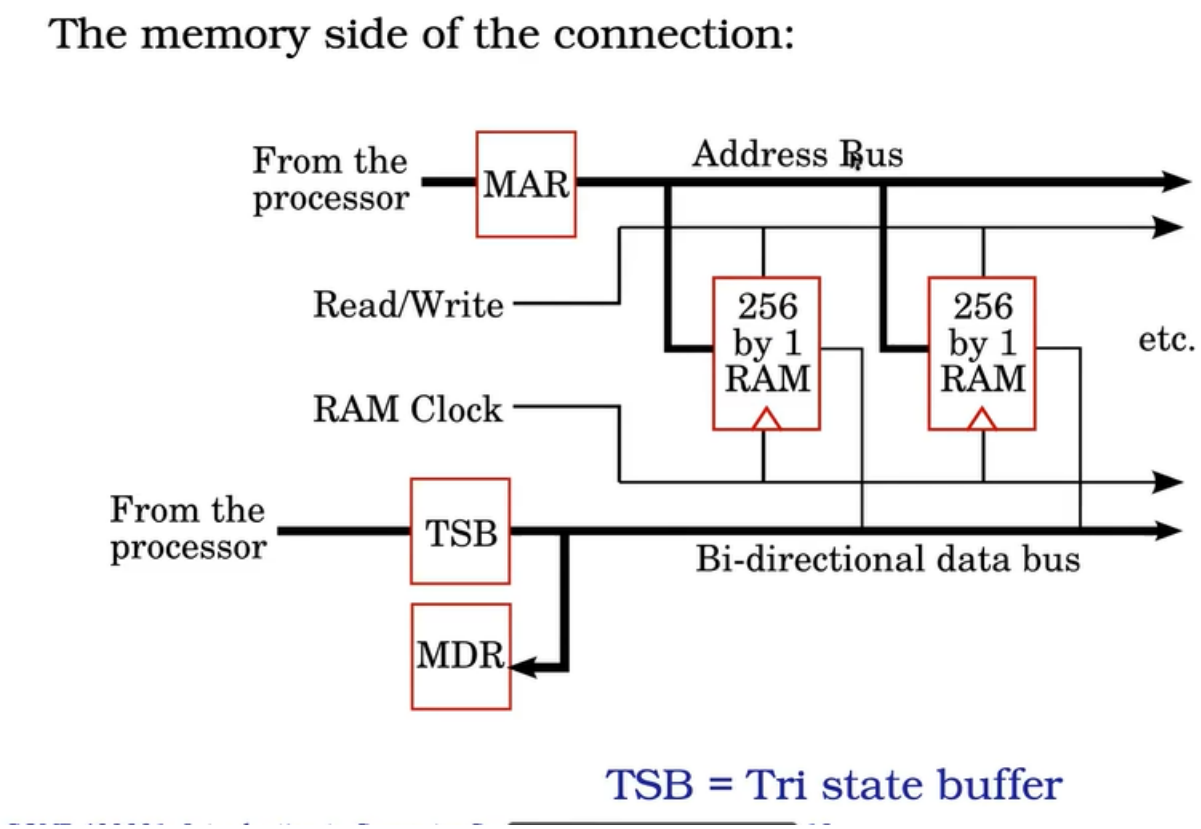
In the square array of one bit memory cells, there will only ever be one cell for which both the row and the column lines are 1.

Each cell is connected to the same read/write line and data line. The data line is connected to the outside through a two way tri state buffer, so that unless the chip is enabled no data can pass either in or out.

This enables us to build external decoders for larger capacity RAMs made up of several banks of single chips.

To use RAM with a processor we need to use two new registers:

- Memory Address Register (MAR)
  - Stores the address in memory to be stored or read
- Memory Data Register (MDR) or Memory Buffer Register (MBR)
  - Stores data read from memory and sometimes data to be written to memory
- These registers are controlled by the processor

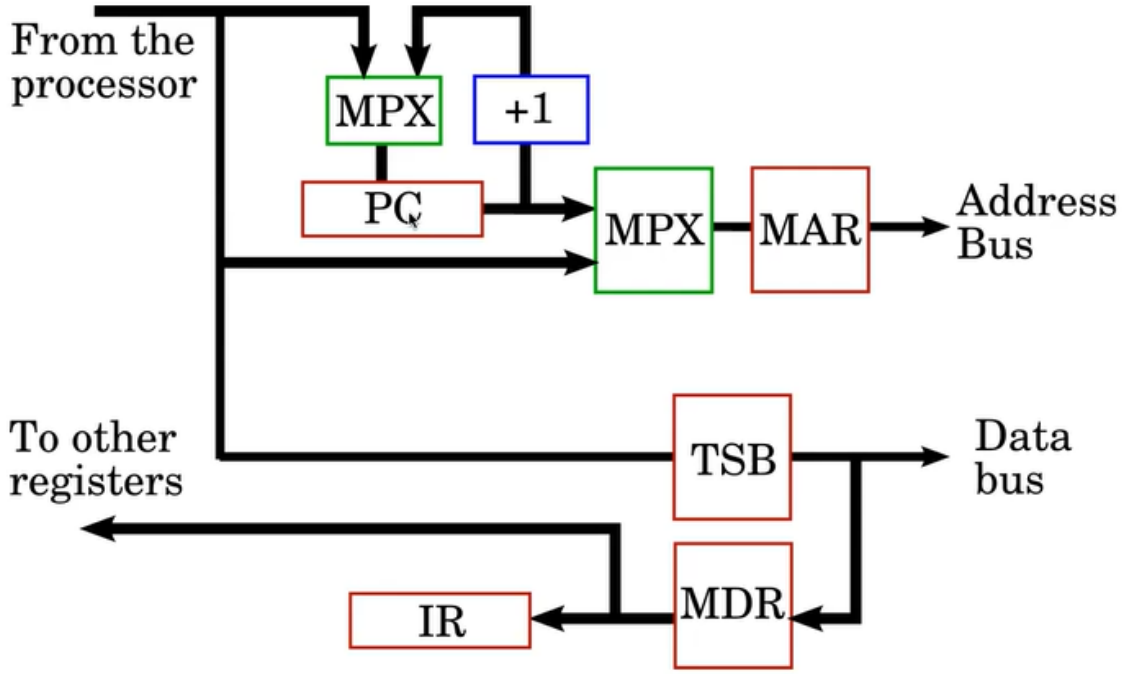


Connecting RAM to a processor

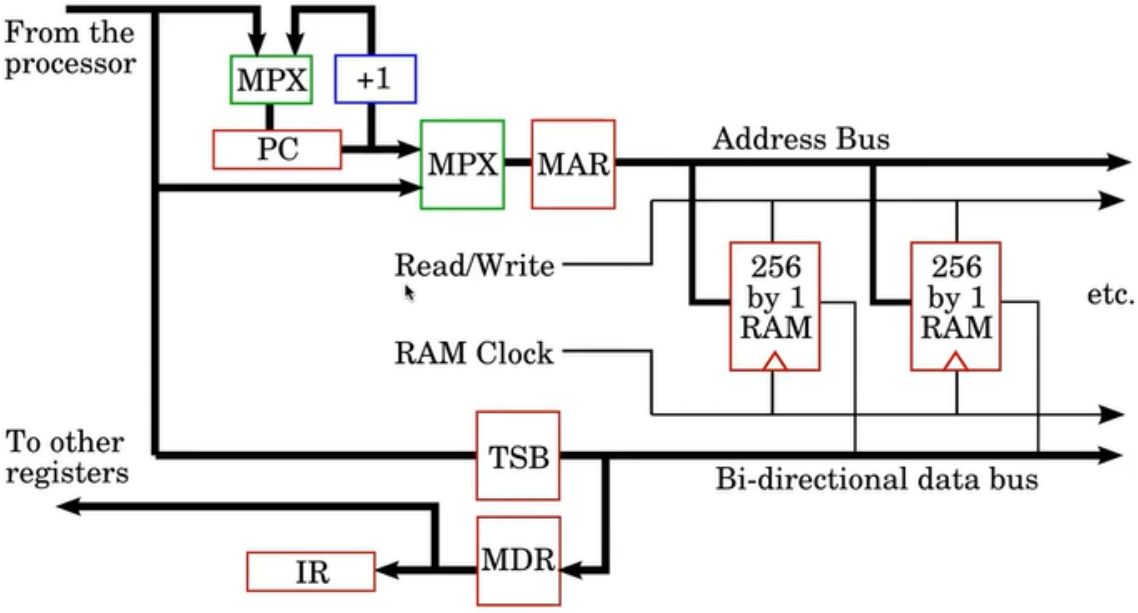
The processor has some additional registers to manage the transfer of data to and from the RAM:

- Program Counter (PC)
  - Stores the address of the next program instruction to be executed
- Instruction Register (IR)
  - Stores the program instruction being executed

The processor side of the connection:



To complete the connection we need some control lines:



The Fetch Cycle

In order to retrieve data from memory we need to go through a number of steps. Each step transfers data from one register to another. The process is called the fetch cycle.

E.g., if we want to get the next program instruction and load it onto the instruction register, we need three steps:

$$MAR \leftarrow PC$$
$$MDR \leftarrow RAM[MAR],\ PC \leftarrow PC+1$$
$$IR \leftarrow MDR$$

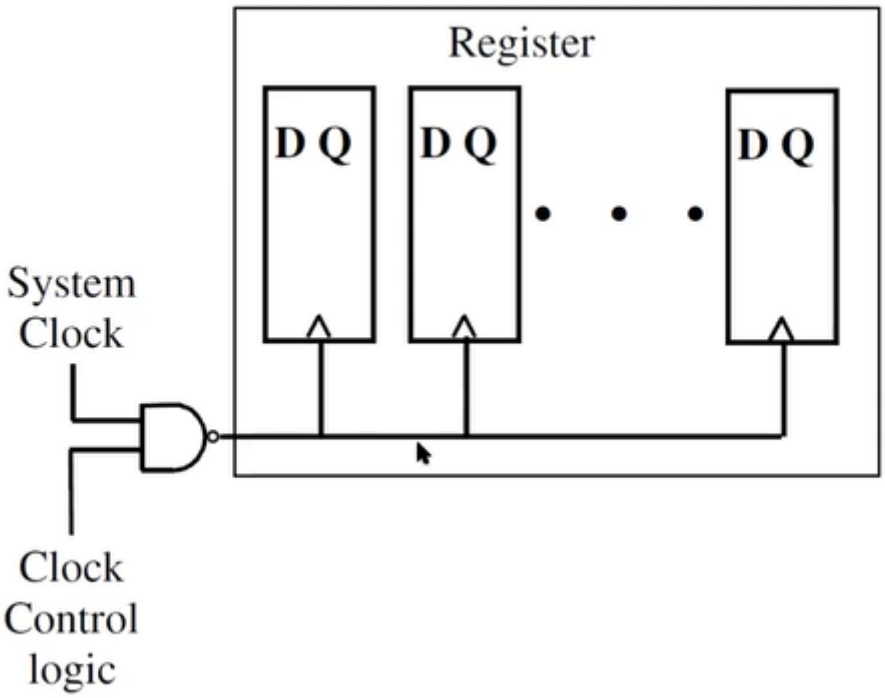
Notice that in the 2nd step, we do 2 register transfers in parallel.

Register Transfers

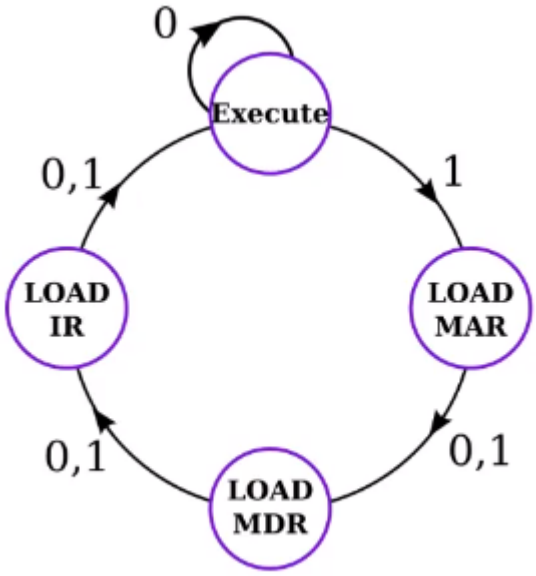
There are two things that must be done in register transfer operations:

1. The multiplexers must be set to establish the required connection paths
2. The registers which are to be loaded must be given a signal (falling edge) on their clock inputs
   These tasks are carried out by a controller which is a synchronous sequential circuit.

Controlling when individual registers are loaded can be done by gating the system clock:



The controller for the fetch cycle is a synchronous sequential circuit. Note that after fetching an instruction from memory the processor will "Execute" it.



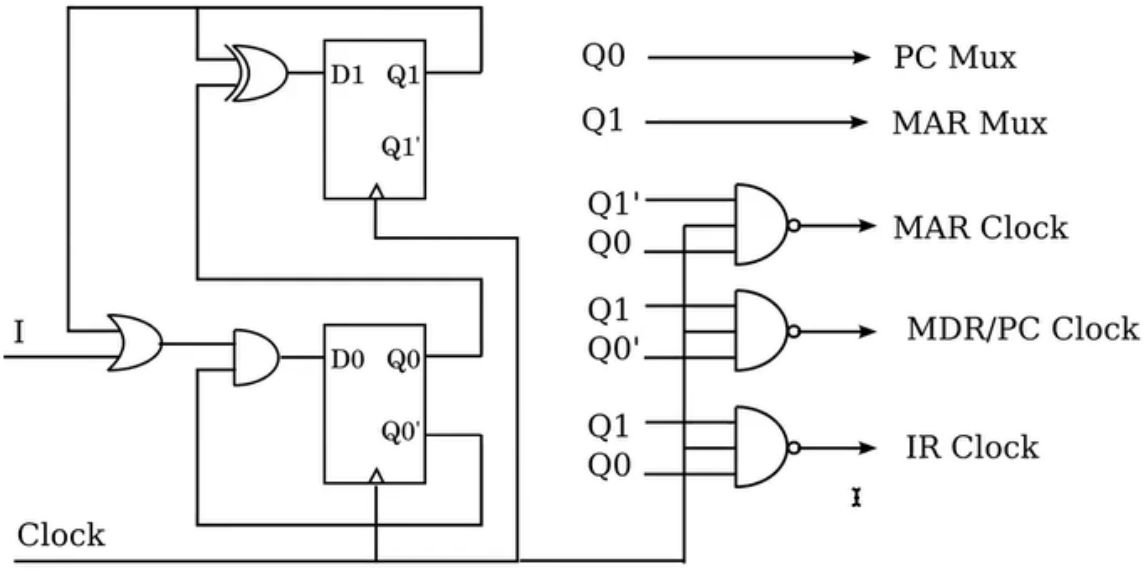
In practice, this will require more than the one state shown here.

The output logic for the Fetch cycle is summarized by the following table:

State	Clock Control				Multiplexer Control	
	MAR	MDR	IR	PC	PC Inp.	MAR Inp.
0. Executing	0	0	0	0	×	×
1. Load MAR	1	0	0	0	×	0
2. Load MDR/PC	0	1	0	1	0	×
3. Load IR	0	0	1	0	×	×

0 sets the MAR input multiplexer to the PC output, and 0 sets the PC input multiplexer to the incrementer output.

The fetch cycle controller is:



Dynamic RAM (DRAM)

For large RAMS (> 1Mbit chips) D-Q flip flops are not used since they are too big. Instead large RAMS use only one transistor and one capacitor for each bit. Capacitor charged = 1; Capacitor uncharged = 0.

The store is not permanent and all the cells storing ones drift to zero in a fraction of a second. A method is employed to restore the capacitor charge regularly. For this reason, circuits of this kind are called dynamic RAM.

Refreshing the capacitor's charges is done when the computer is not accessing the memory. A memory controller must tell the DRAM to refresh itself periodically...This overhead is very low.