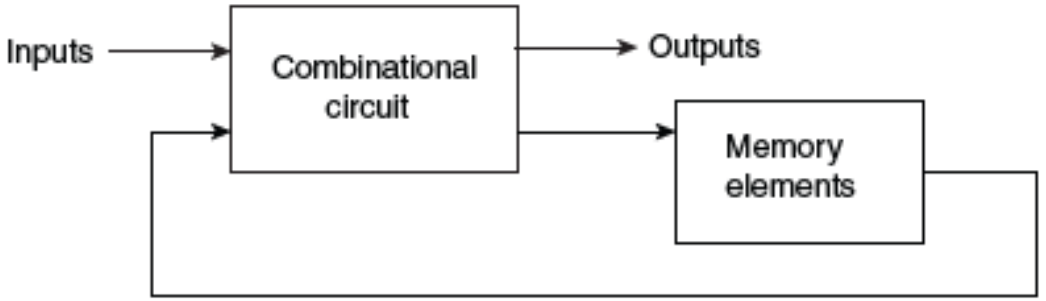
In this topic, we'll learn

Sequential Circuits

- Flip Flops
- State diagrams
- Counters

Introduction

- Sequential logic: Systems with combinational circuits and memory elements are described.
- Feedback path.
- Memory elements
  - Stores binary information
  - Binary information defines the state of the sequential circuit.



- External i/ps plus state of the memory elements determine the o/ps and the next state of the memory element.
- Sequential circuit is specified by a time sequence of i/ps, o/ps and internal states.

Synchronous vs Asynchronous Sequential Circuits (based on timing)

Synchronous

- Behavior can be defined from the knowledge of its signals at discrete instants of time.
- Synchronization by master-clock generator.
  - Produces periodic train of clock pulses.
  - Distributed throughout the system.
  - Memory systems are affected only with arrival of the synchronization pulse (AND gates)

Asynchronous

- Behavior depends upon
  - the order in which its i/p signals change
  - Can be affected at any instant of time.
- Memory devices used are time-delay devices.
- Unreliable due to variation in delays of the i/ps.

Clocked Sequential circuits

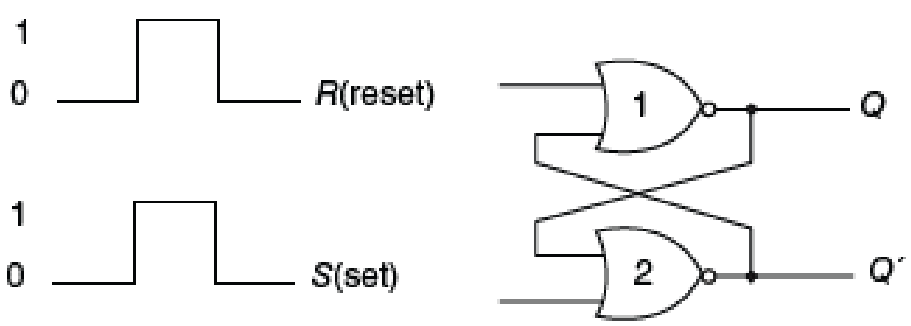
- Synchronous Sequential circuits that use clock pulses in the i/ps of the memory elements: Clocked Sequential circuits.
- Most frequent.
- Do not have instability problems.
- Timing is broken into independent discrete steps.
- Sequential circuits we discuss are clocked sequential circuits.
- Memory elements used are called flip-flops.
  - Binary cells capable of storing one bit of information.
  - 2 o/ps: normal and complement values.
  - Various types of flip-flops: based on the entry of binary information.

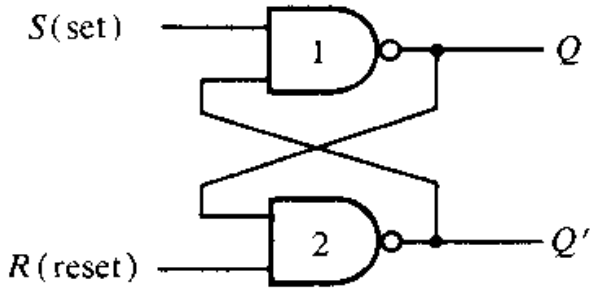
Flip-Flops

- Maintain a binary state indefinitely (as long as power is delivered to the circuit)
- States are switched when directed by i/p signals.
- Different types of FF
  - Number of i/ps
  - Manner in which i/ps affect the binary state.

Basic Flip-Flop circuit

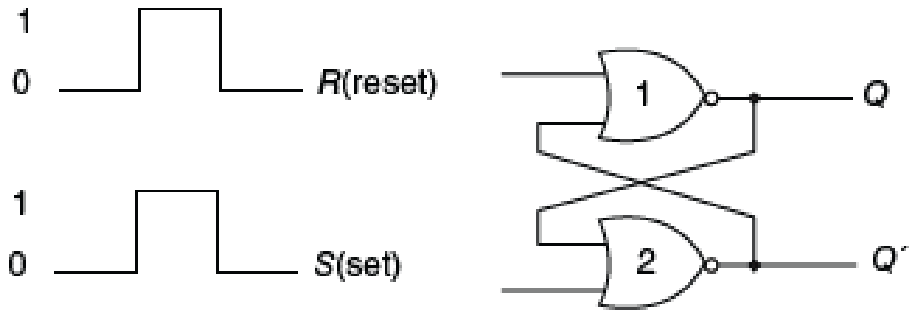
- Constructed from 2 NAND or NOR gates.
  - more complicated types can be built from these basic FFs.
- Cross-coupled connection from the o/p of one gate to the i/p of the other gate: Feedback
  - Asynchronous sequential circuits
- 2 o/ps: Q and Q', 2 i/ps: set and reset
- Binary state of FF is the value of the normal o/p Q.





NOR Flip-Flop circuit

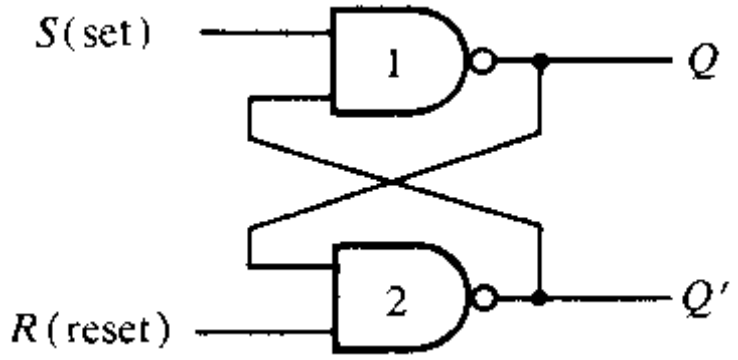
- Set: S=1, R=0
- Reset: S=0, R=1
- Memory: S=R=0
- Undefined: S=R=1



S	R	Q	Q'	
1	0	1	0	
0	0	1	0	(after $S = 1, R = 0$)
0	1	0	1	
0	0	0	1	(after $S = 0, R = 1$)
1	1	0	0	

NAND Flip-Flop circuit

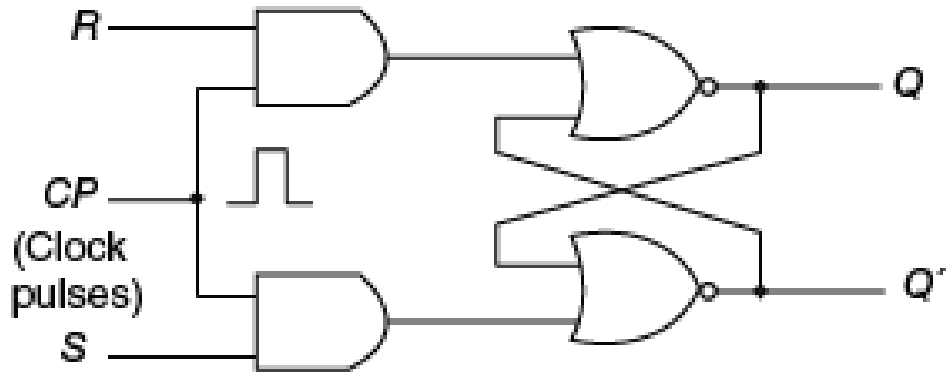
- Set: S=0, R=1
- Reset: S=1, R=0;
- Undefined: S=R=0
- Memory: S= R=1

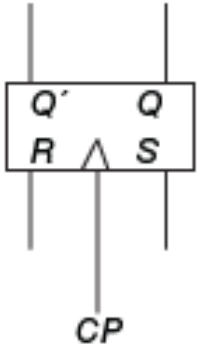


S	R	Q	Q'	
1	0	0	1	
1	1	0	1	(after $S = 1, R = 0$)
0	1	1	0	
1	1	1	0	(after $S = 0, R = 1$)
0	0	1	1	

Clocked *RS* Flip-Flop

- Graphic Symbol
  - triangle is a symbol for a *dynamic indicator*
  - Indicates: FF responds to an i/p clock transition from 0 to 1.
  - State of the FF is determined by Q.

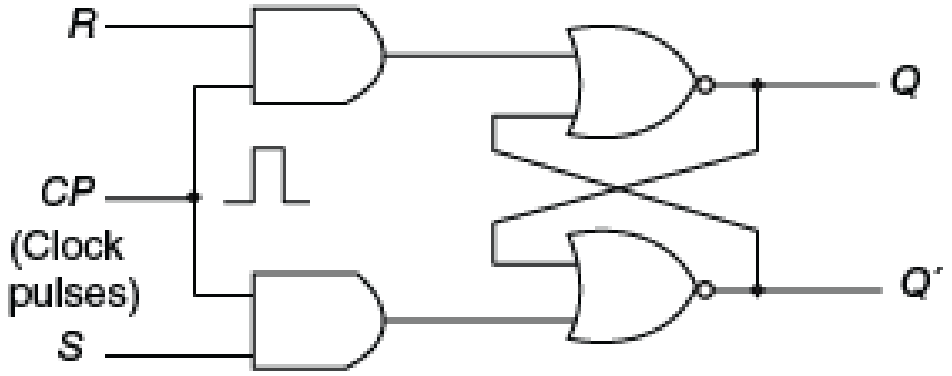




Clocked *RS* Flip-Flop

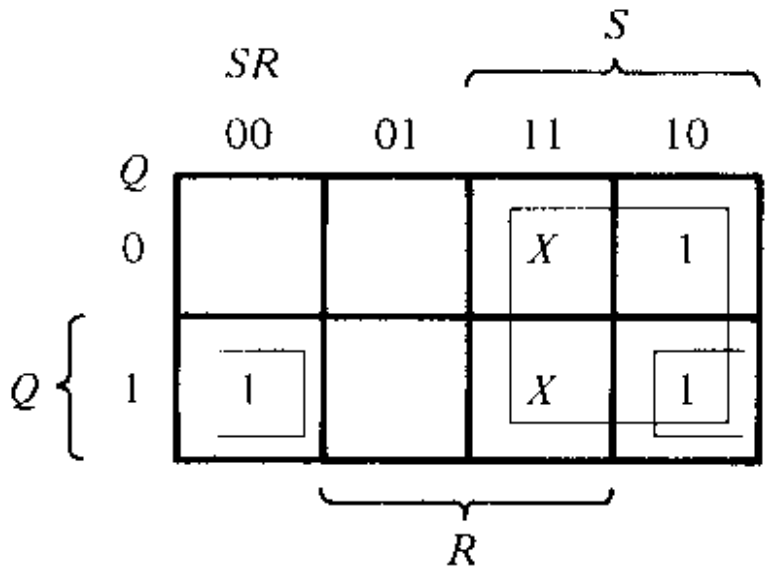
- Characteristic table

Q	S	R	$Q(t+1)$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	indeterminate
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	indeterminate



Clocked *RS* Flip-Flop

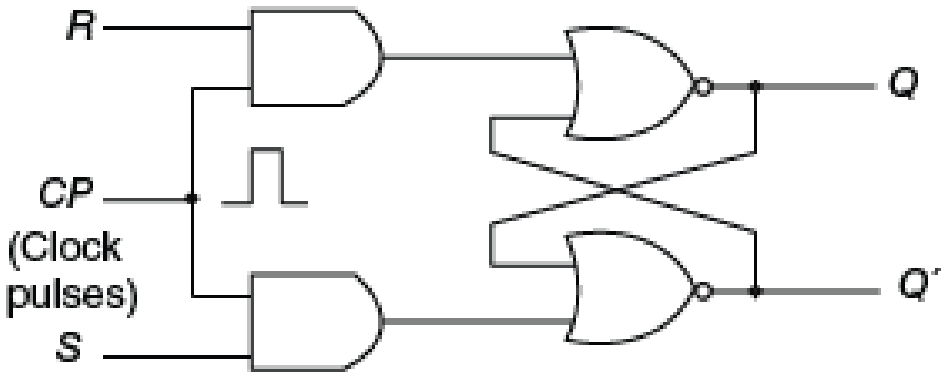
- Characteristic equation
  - specifies the value of the next state as a function of the present state and the inputs.
  - an algebraic expression for the binary information of the characteristic table.



$$Q(t+1) = S + R'Q$$
$$SR = 0$$

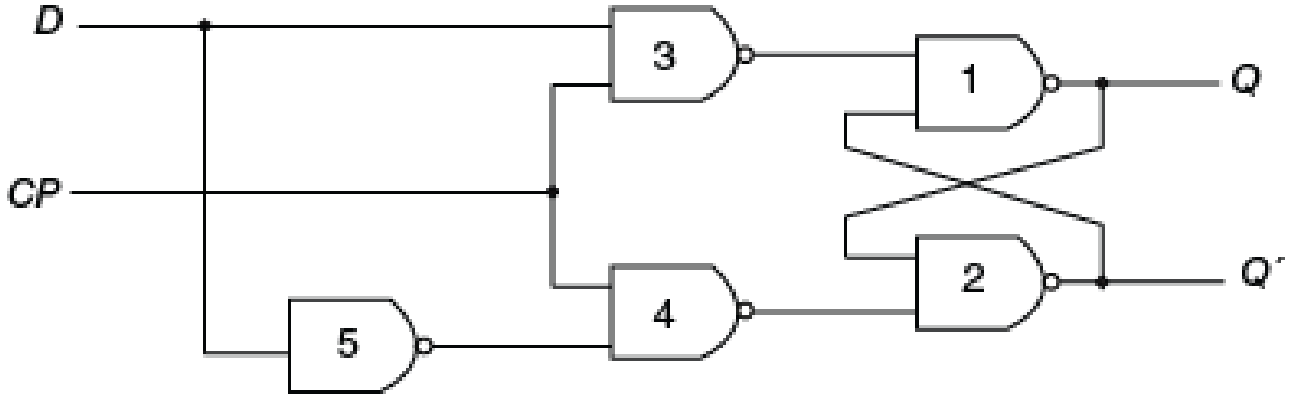
Clocked *RS* Flip-Flop

- Characteristic equation
  - specifies the value of the next state as a function of the present state and the inputs.
  - an algebraic expression for the binary information of the characteristic table.
  - two indeterminate states are marked by *X*'s.
  - $SR = 0$: both *S* and *R* cannot equal 1 simultaneously



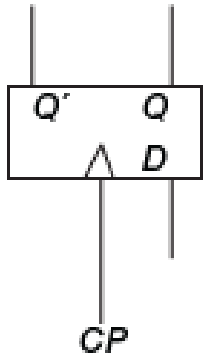
D Flip-Flop (transfer “data”)

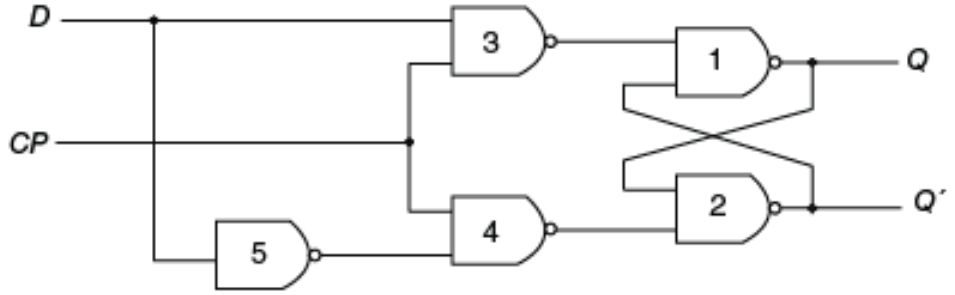
- Modification of the clocked *RS* flip-flop
  - *RS* flip-flop with an inverter in the *R* input.
  - Also called *gated D-latch.*
  - *CP* input is often given the variable designation *G* (*for gate*).



D Flip-Flop (transfer “data”)

- Modification of the clocked *RS* flip-flop
  - *RS* flip-flop with an inverter in the *R* input.
  - Also called *gated D-latch.*
  - *CP* input is often given the variable designation *G* (*for gate*).
- Graphic Symbol

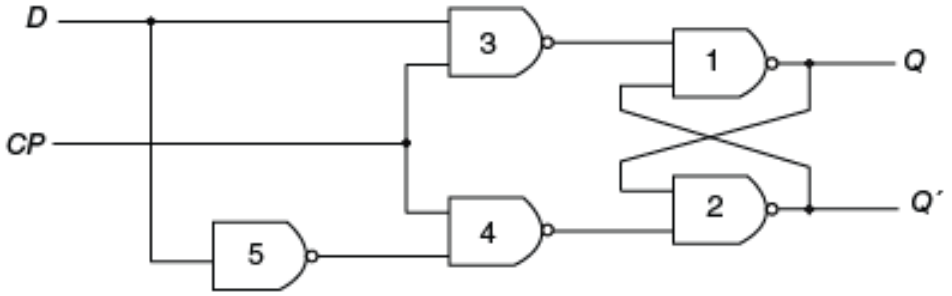




D Flip-Flop (transfer “data”)

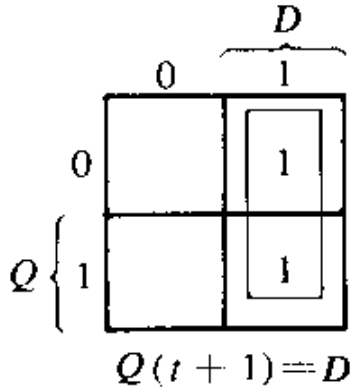
- Modification of the clocked *RS* flip-flop
  - *RS* flip-flop with an inverter in the *R* input.
  - Also called *gated D-latch.*
  - *CP* input is often given the variable designation *G* (*for gate*).
- Characteristic table

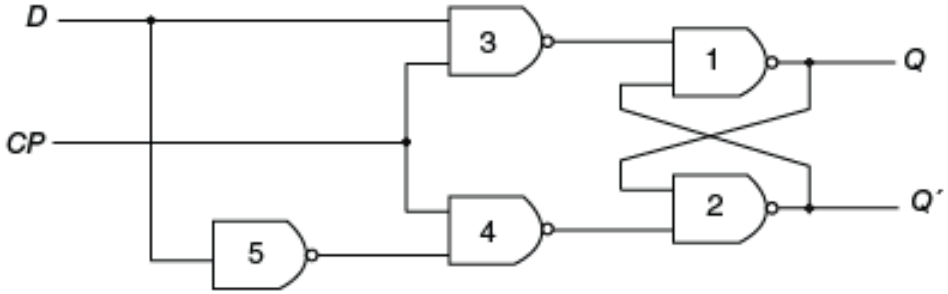
Q	D	$Q(t+1)$
0	0	0
0	1	1
1	0	0
1	1	1



D Flip-Flop (transfer “data”)

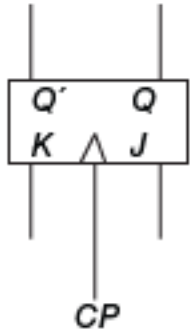
- Modification of the clocked *RS* flip-flop
  - *RS* flip-flop with an inverter in the *R* input.
  - Also called *gated D-latch.*
  - *CP* input is often given the variable designation *G* (*for gate*).
- Characteristic Equation

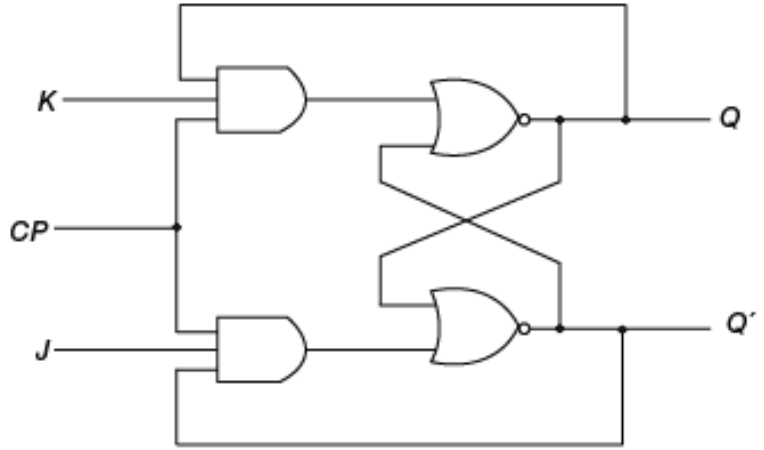




JK Flip-Flop

- Refinement of the *RS* flip-flop
  - indeterminate state of the *RS* type is defined in JK.

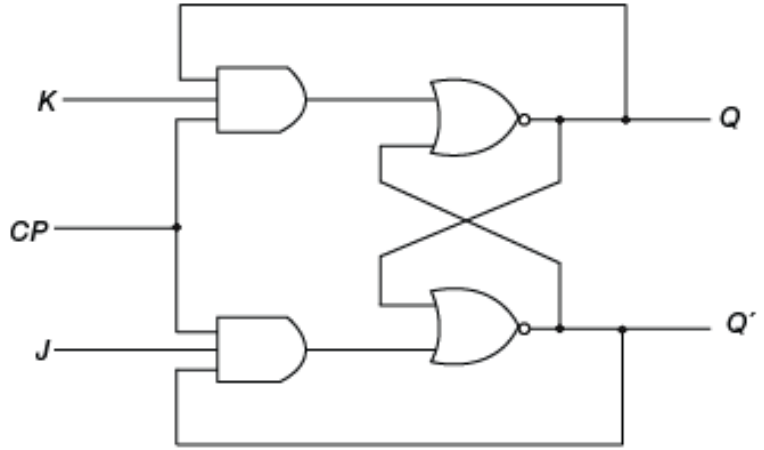




JK Flip-Flop

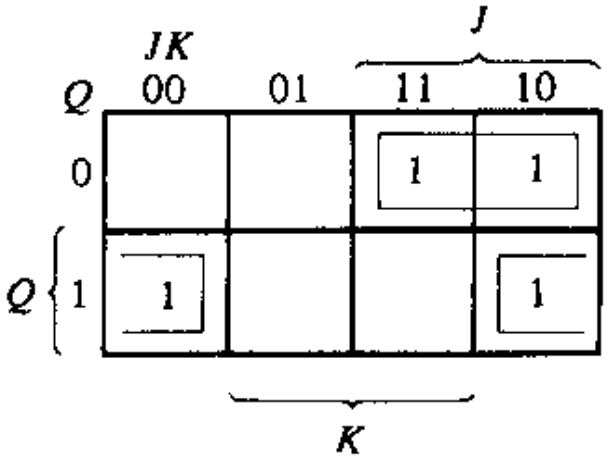
- Refinement of the *RS* flip-flop
  - indeterminate state of the *RS* type is defined in JK.
- Characteristic Table

Q	J	K	$Q(t+1)$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

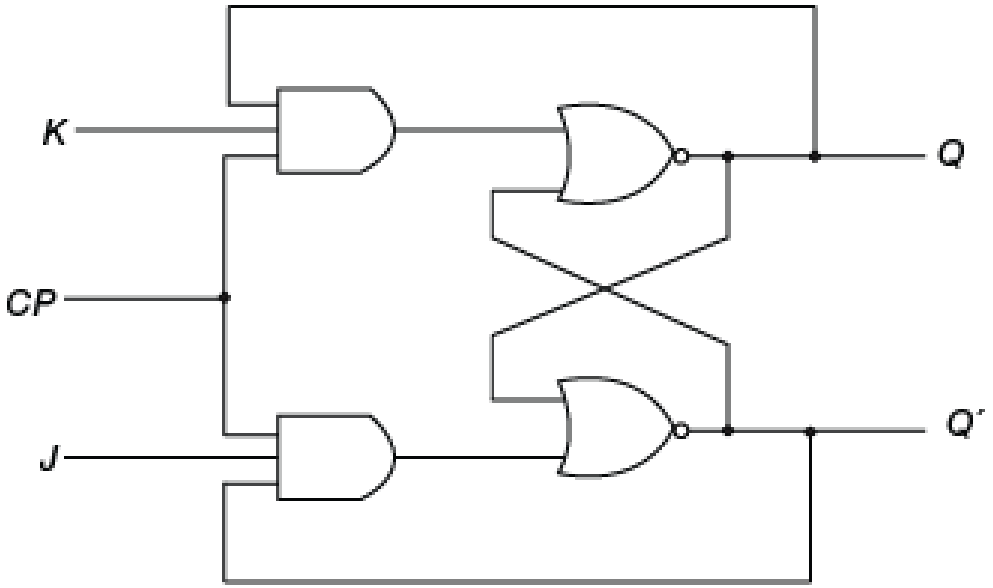


JK Flip-Flop

- Refinement of the *RS* flip-flop
  - indeterminate state of the *RS* type is defined in JK.
- Characteristic equation

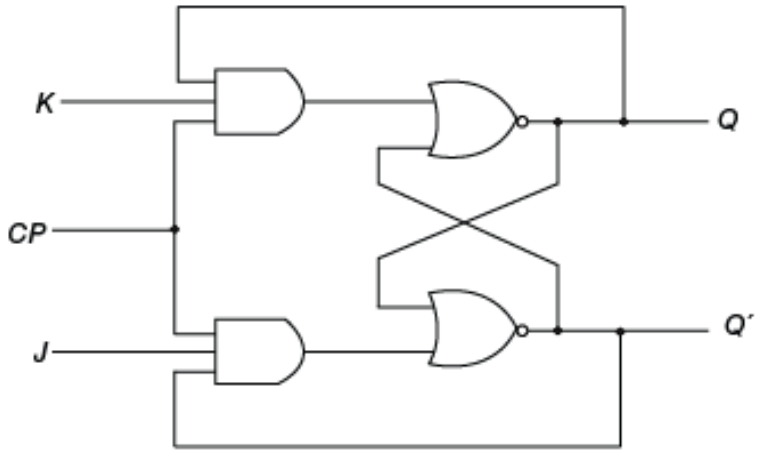


$$Q(t+1) = JQ' + K'Q$$

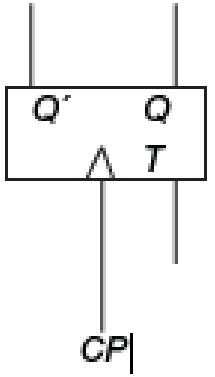


JK Flip-Flop

- Refinement of the *RS* flip-flop
  - indeterminate state of the *RS* type is defined in JK.
- If CP=1 and J=K=1
  - Repeated and continuous transitions of o/ps.
  - To avoid this
    - CP must be shorter than the propagation delay through the FF.

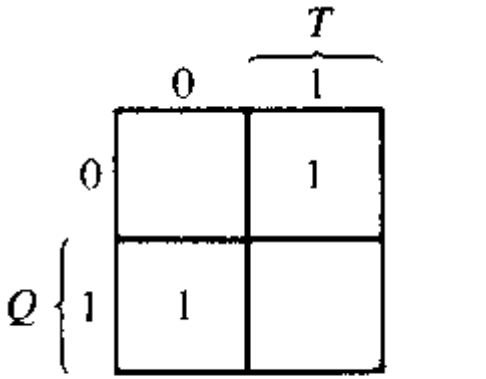


T Flip-Flop

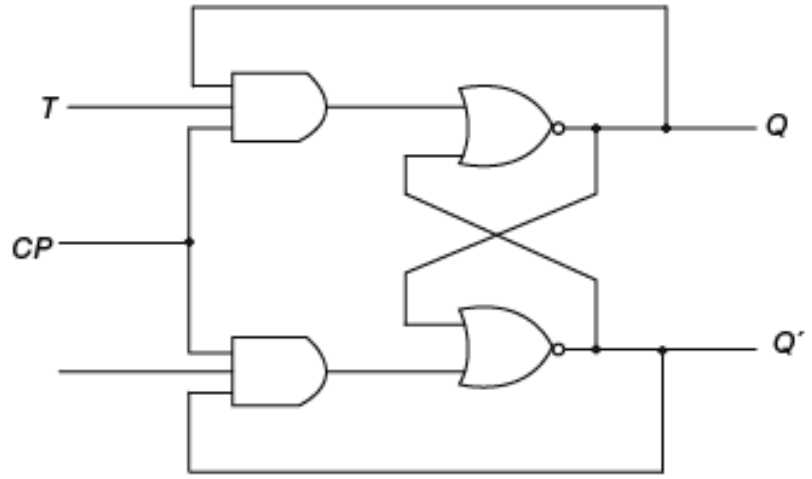


- Single i/p version of JK FF.
  - Both J and K are tied together.
  - T stands for “toggle”, when T=1 and CP=1.
- Obtain the Characteristic Table and Characteristic equation.

Q	T	$Q(t+1)$
0	0	0
0	1	1
1	0	1
1	1	0



$$Q(t+1) = TQ' + T'Q$$



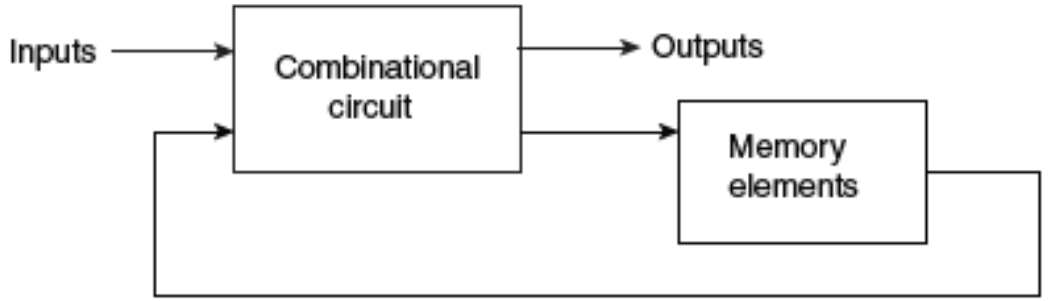
Different conventions used
Latch vs Flip-Flop

- Latch is level-triggered.
- Flip-Flop is edge-triggered.

Triggering of Flip-Flops

- Switch in the state of a FF by momentary change in the i/p signal
  - Momentary change: Trigger
- For Latches (Asynchronous FF) triggering is by change of signal level.
  - Level should return to its initial value (0 in NOR and 1 in NAND) before second trigger.
- Flip-Flops (Clocked FF) are triggered by pulses.
  - Pulse start from an initial value of 0, goes briefly to 1, and returns to its initial value of 0.
  - Time interval from the application of pulse until the o/p transition is **critical.**

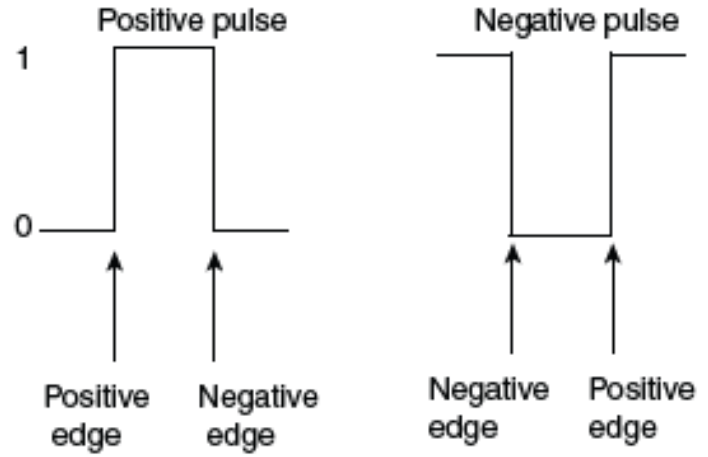
Feedback path and instability



- Instability
  - If o/ps of FFs are changing while the o/ps of the combinational circuits (i/ps to FFs) are being sampled by the clock pulse.
- Can be prevented
  - If o/ps of FFs do not start changing until pulse i/p returns to 0.
  - Signal propagation delay of a FF from the i/p to o/p should be greater than the pulse duration. (Difficult to control)
  - Include a physical unit for the delay or
  - Make FF sensitive to pulse transition.

Definition of Clock pulse transition

- Positive pulse: 1 during the occurrence of pulse. 0 otherwise.
- 0 to 1: Positive edge
- 1 to 0 negative edge



Multiple-transition problem

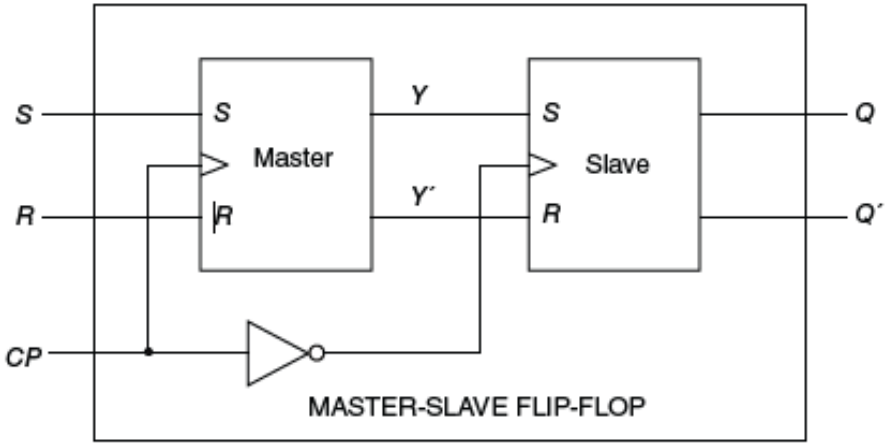
- Clocked flip-flops introduced
  - Triggered during the positive edge of the pulse.
  - State transition starts immediately after pulse becomes 1.
  - New state of the FF may appear at the o/p while the pulse is still 1.
  - FF will start responding to these new values.
  - A new o/p may occur.
  - Hence, o/p of 1 FF cannot be applied to the i/p of another FF when both are triggered by the same clock pulse.
- Can be eliminated if FF respond to edge transition only.

Capacitive coupling

- RC circuit is inserted in the clock i/p of the FF.
  - Generates a spike in response to momentary change in i/p.
  - Positive spike: At positive Edge; Negative spike: At negative Edge
- Edge triggering: By designing the FF to respond to one spike and neglect the other.

Master-Slave Flip-Flop

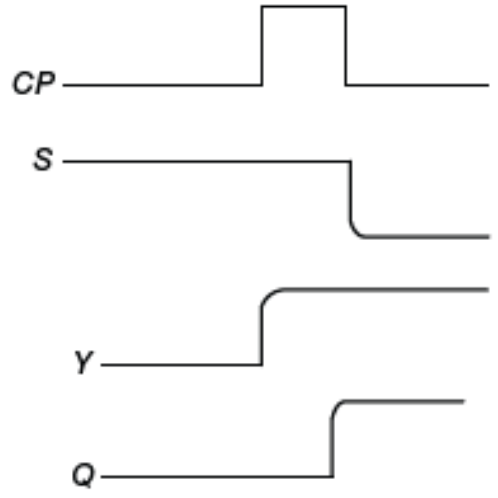
- Constructed from two separate FFs.
  - One master and other acts as slave.
- When CP=0
  - Slave is enabled.
  - Q=Y
  - Master is disabled.
- When CP=1
  - Master is enabled.
  - Information at S and R i/ps is transmitted to Y.
  - Slave is disabled.
-



- When CP returns to 0
  - Slave goes to the same state as the master.

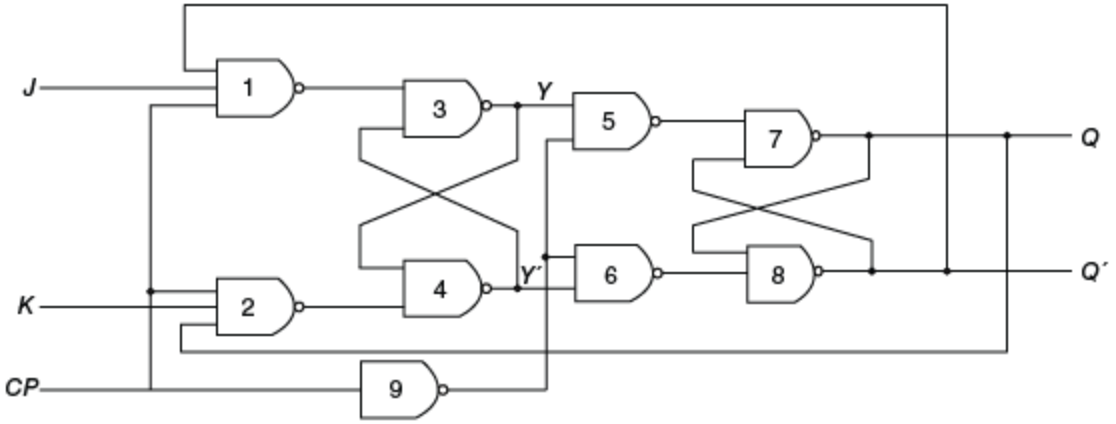
Timing relationships in a master-slave flip-flop

- Initially FF is cleared
  - Y=Q=0.
- *S* input can be changed at the same time that the pulse goes through its negative edge transition.
- Once the *CP* reaches 0, the master is disabled.
  - possible to use the same clock pulse to switch
    - output of the flip-flop
    - input information.
- State changes at the negative edge transition of the clock pulse.



Clocked master-slave *JK* flip-flop

- Gates 1 to 4: Master FF.
  - 5 to 8: Slave FF



Cascading of many Master-Slave FFs

- When pulse is 1 all the masters (internal to the FF) are enabled
  - O/p of the FFs are not affected.
- After the clock returns to 0.
  - Slaves are enabled.
  - O/ps of some of the FFs are changed.
  - None of the masters are affected by these changes.

Edge-Triggered Flip-Flop

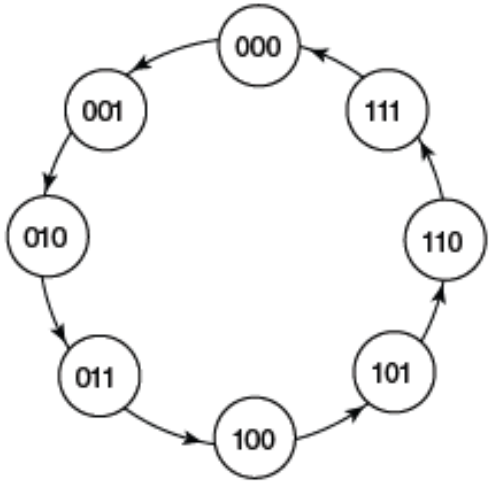
- Output transitions occur at a specific level of the clock pulse
  - When the pulse input level exceeds this threshold level, i/ps are locked out.
  - Could be positive or negative edge triggered.

Design of Counters

- Counters: Sequential circuits which undergo **prescribed sequence of states** upon application of **i/p pulse**.
- I/p pulse: Called **count pulse**.
  - clock or from an external source.
  - Prescribed time or random.
- Sequence: binary (simplest and straightforward) or any other
- Used in all equipment with digital logic.
  - Number of occurrences.
  - Timing sequences to control operations.

Binary Counters

- Simplest and straight forward.
- n-bit counter: **n FFs** and count from **0 to 2^n-1**.



- FF count **repeats**. Goes to 000 after 111.
- i/p and o/p values not shown.
- Clocked sequential circuits: State transitions during clock pulses. **Not shown** explicitly.
- Only i/p: **Count pulse**.
- O/ps: Specified by the **present states** of FFs.
- Next state
  - **Depends** only on the **present state**.
  - Transitions during clock pulses.
  - **Completely specified** by the count sequence.

Excitation Table for 3-bit counter

Table 6-12 Excitation table for a 3-bit binary counter

Count sequence			Flip-flop inputs		
A_2	A_1	A_0	TA_2	TA_1	TA_0
0	0	0	0	0	1
0	0	1	0	1	1
0	1	0	0	0	1
0	1	1	1	1	1
1	0	0	0	0	1
1	0	1	0	1	1
1	1	0	0	0	1
1	1	1	1	1	1

- **Next number** represents the **next state**.
- Count sequence: Provides all information to design the circuit.
- Follows the **same procedure**.
- Excitation obtained **directly** from the count sequence.
- **Binary counters** most **effectively constructed** by **T FFs**.
- **Last row** compared with the first count **000,** its **next state**.

Memory Unit

- Operation register:
  - capable of **storing binary information** in its flip-flops
  - has **combinational gates** for **data-processing** tasks.
- Storage register:
  - **temporary storage** of binary information.
- *Memory Unit:* Collection of **storage registers** together with the **associated circuits** needed to **transfer** information in and out of the registers.
- *Memory registers:* **Storage registers** in a memory unit.
- **Most** of the **registers** in a digital computer are **memory registers**.
- Comparatively **few operational registers** are found in the processor unit.
- When data processing takes place:
  - **Information** from **selected registers** in memory unit is first **transferred** to **operational registers** in the processor unit.
  - **Intermediate** and **final** results obtained in the operational registers are **transferred back** to **selected memory registers**.

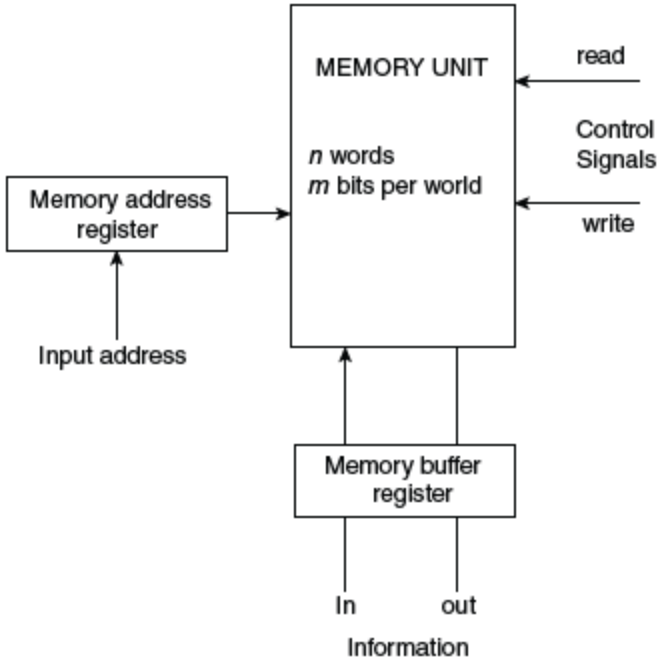
Basic properties of binary cells of Registers

- **Reliable two-state** property for binary representation.
- **Small** in size.
- **Cost per bit** of storage should be as **low** as possible.
- The **time of access** to a memory register should be **reasonably fast**.
- Examples of memory unit components
  - magnetic cores, semiconductor ICs, and magnetic surfaces on tapes, drums, or disks.

Memory Word

- *Words:* Memory unit stores binary information in **groups**.
- Each **word stored** in a **memory register**.
- Word is an **entity of *n* bits** that moves in and out of storage as a unit.
- A memory word may represent
  - an **operand**, an **instruction**, a group of **alphanumeric characters**, or any **binary-coded information**.

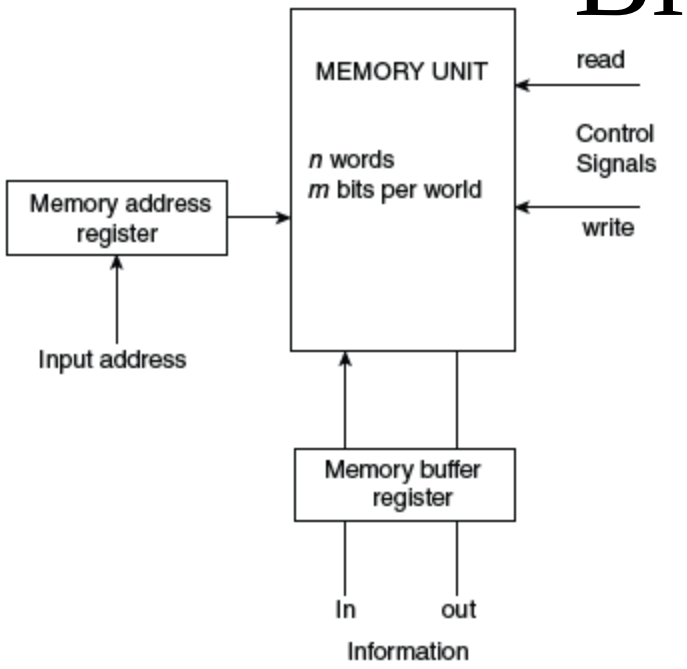
Block Diagram



- communication between a memory unit and its environment is achieved through
  - **two control signals**
  - **two external registers**.
- The control signals specify the direction of transfer required whether
  - a word is to be stored in a memory register (**Write**)
  - whether a word previously stored is to be transferred out of a memory register. (**Read**)

- **Memory Address Register (MAR)**: specifies the particular memory register chosen.
- **Memory Buffer Register (MBR)**: specifies the particular bit configuration of the word.

Block Diagram

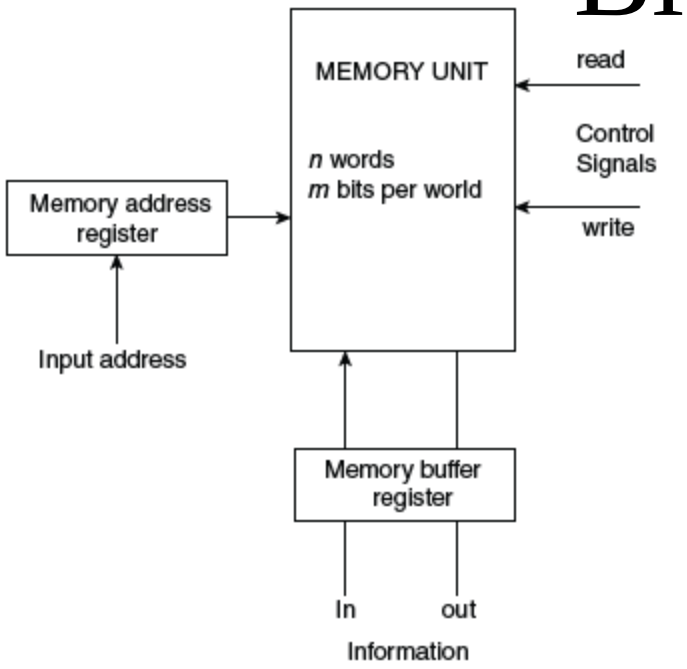


- Internal circuits of the memory unit
  - accept this address from the register
  - open the paths needed to select the word called.
- An address register with *n* bits: specify up to 2^n memory words.

- Memory Address Register (MAR)
  - Specifies the **memory word selected**.
- Each word in memory
  - assigned a number **identification** starting from 0 to maximum number of words
- *Address of a word:*
  - transferred to MAR for communication.

Computer memory units can range from 1024 words (**10 bit-MAR**) , to 1,048,576 (**20 bit- MAR**) = 2^{20} words,

Block Diagram



- Memory Buffer Register (MBR)
  - **information transfer** to and from registers in memory and external environment
- **Write**
  - internal control interprets: the **contents of the MBR** to be stored in a memory register.

- Read:
  - internal control **sends** word from a memory register into the MBR.

- Contents of MAR specify particular memory register **referenced** for writing or reading

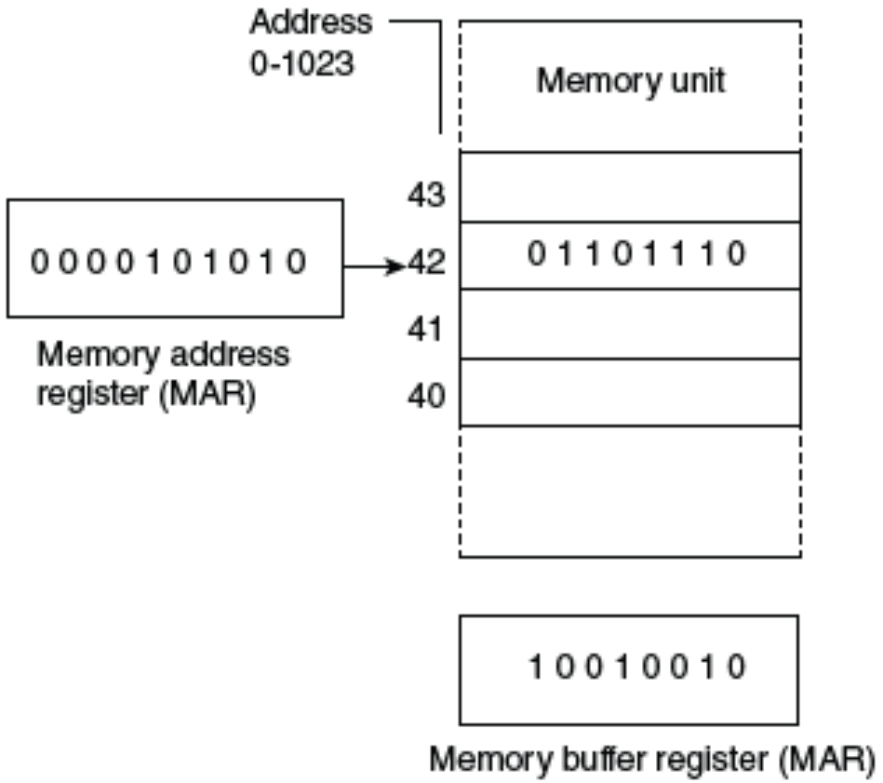
1024 words with eight bits per word.

- In MAR, we need an address of 10 bits ($2^{10} = 1024$)
- Hence10 no. of FFs are required

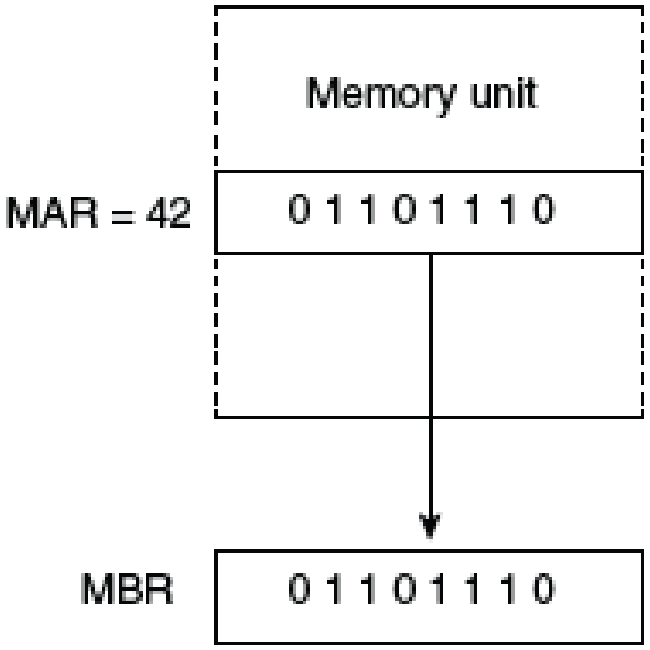
- Similarly, MBR will have 8 no. of FFs

- Memory unit will have 1024 no. of Registers assigned with address 0 to 1023.

Example: Initial conditions in the three registers

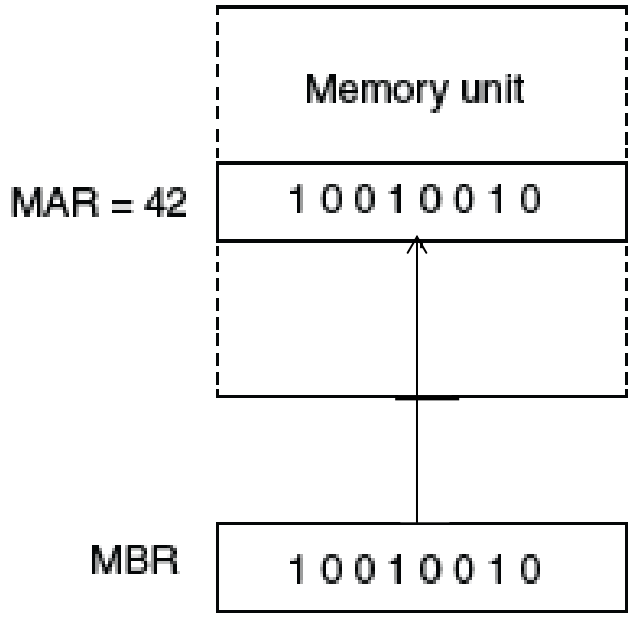


Read operation



- The **sequence of operations** needed to communicate with memory unit for transferring a word out to the MBR is:
- 1. **Transfer** the **address bits** of the selected word into **MAR**.
- 2. Activate the ***read*** control input.
- The binary information stored in memory register 42 is transferred into MBR.

Write Operation

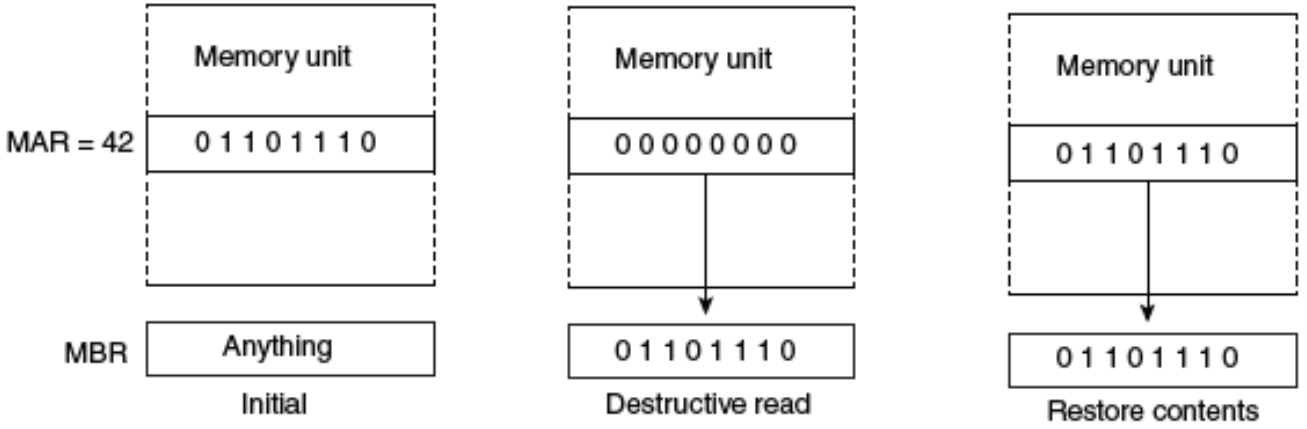


- The **sequence of operations** needed to store a new word into memory is:
- 1. Transfer the **address bits** of the selected word into **MAR**.
- 2. Transfer the **data bits** s the word into **MBR**.
- 3. Activate the ***write*** control input.

Types of memory

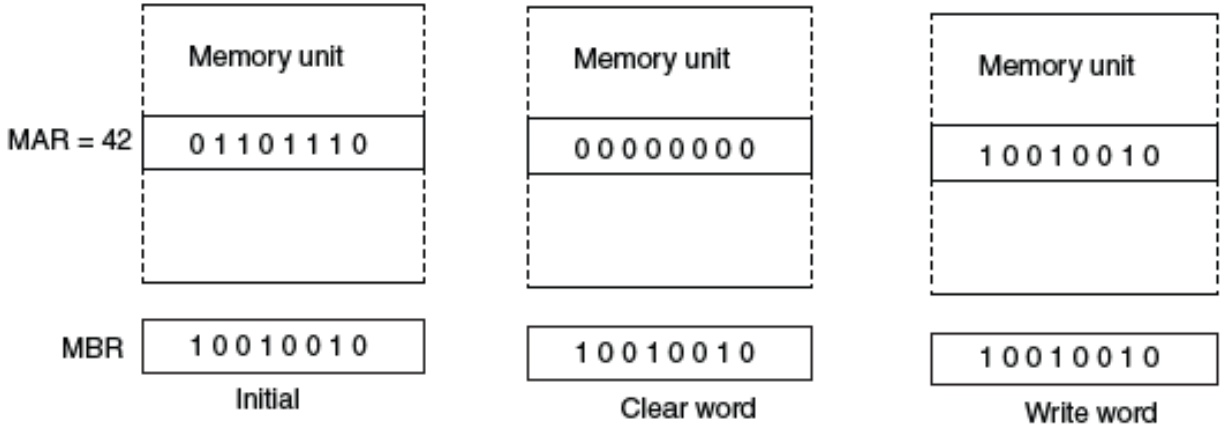
- Using **Semiconductor IC**:
  - **Non destructive** Read.
  - **Retain** their information after Read.
  - **No loss** of information.
- Magnetic core:
  - **Destructive** Read-out.
  - **Losses** the stored information during Read.
  - must provide **additional control** functions **to restore** the word into the memory register

Read operation in Magnetic core memory



- During Restore **contents** of **MAR** and **MBR** must be **unchanged**.

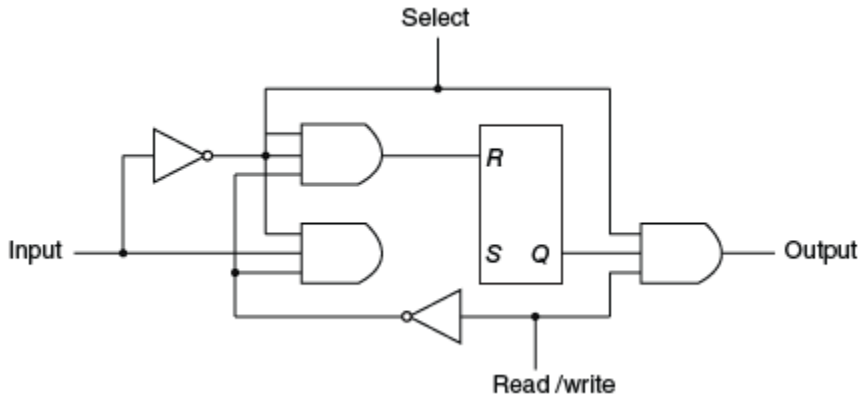
Write operation in Magnetic core memory



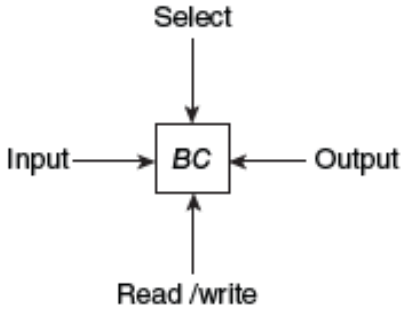
- **MAR** must **not change** during operation
  - To **ensure** that the **same selected word** cleared receives **information**.

- Magnetic-core memory requires **2 half-cycles** for reading or writing.
- **Time** to go through **both cycles**: **Memory-cycle** time.

Integrated Circuits memory (RAM)

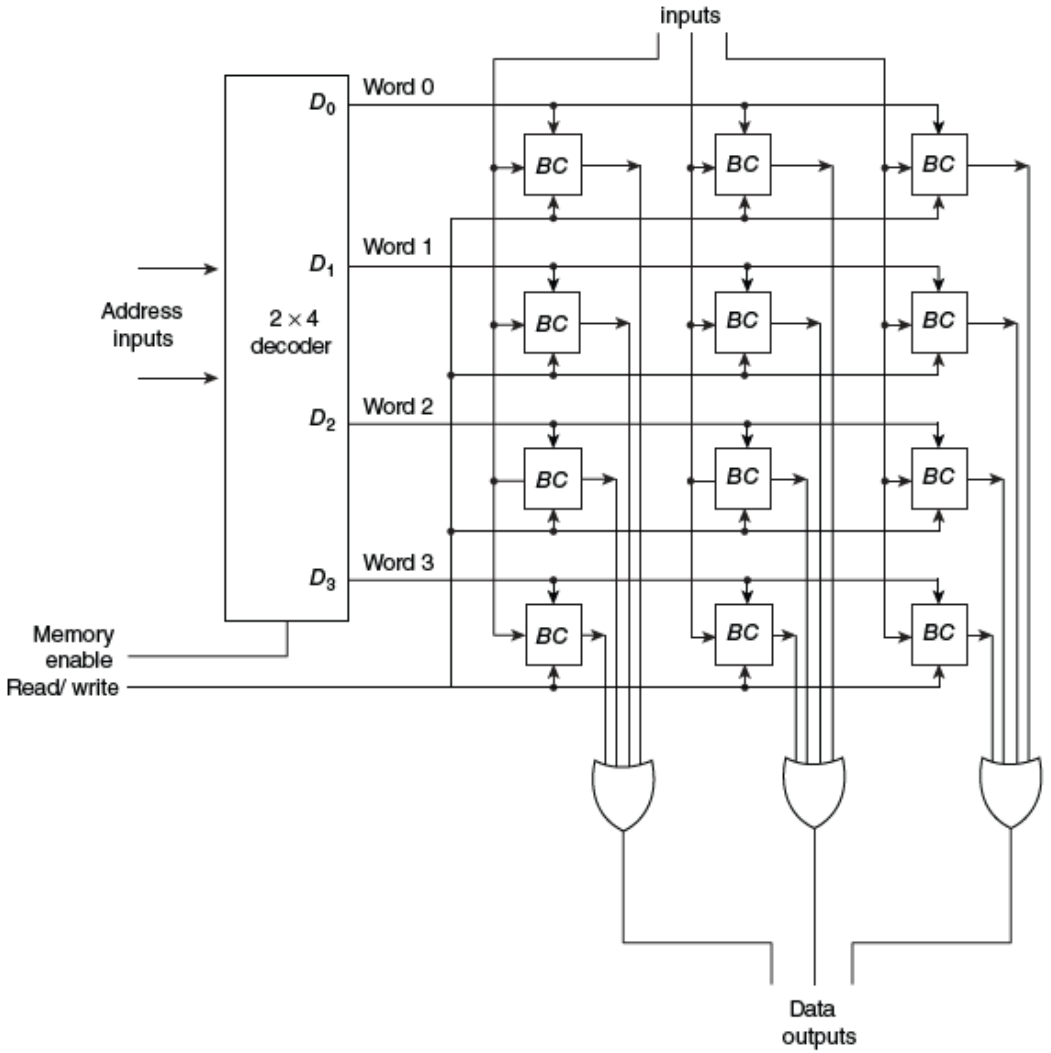


- **Binary storage cell**: Basic building block of memory unit.
- Must be **very small**
  - To pack as many cells as possible.
- **3 i/ps** and **1 o/p**.



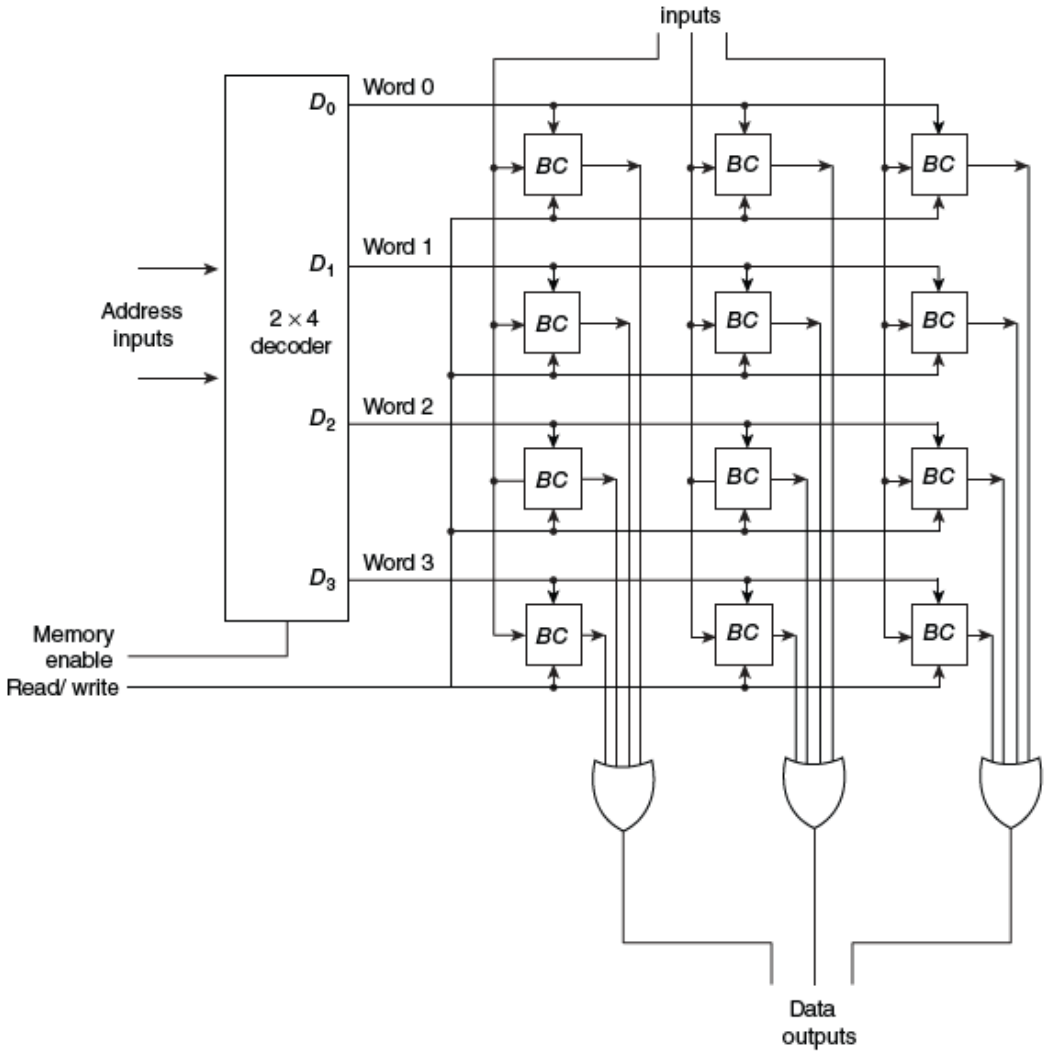
- **Select i/p: Enables** cell for reading or writing.
- **Read/write**:
  - 1 for Read.
  - 0 for Write.
- FF
  - operates **w/o** clock pulses.
  - **Store** information bit

Integrated Circuits memory unit (RAM)



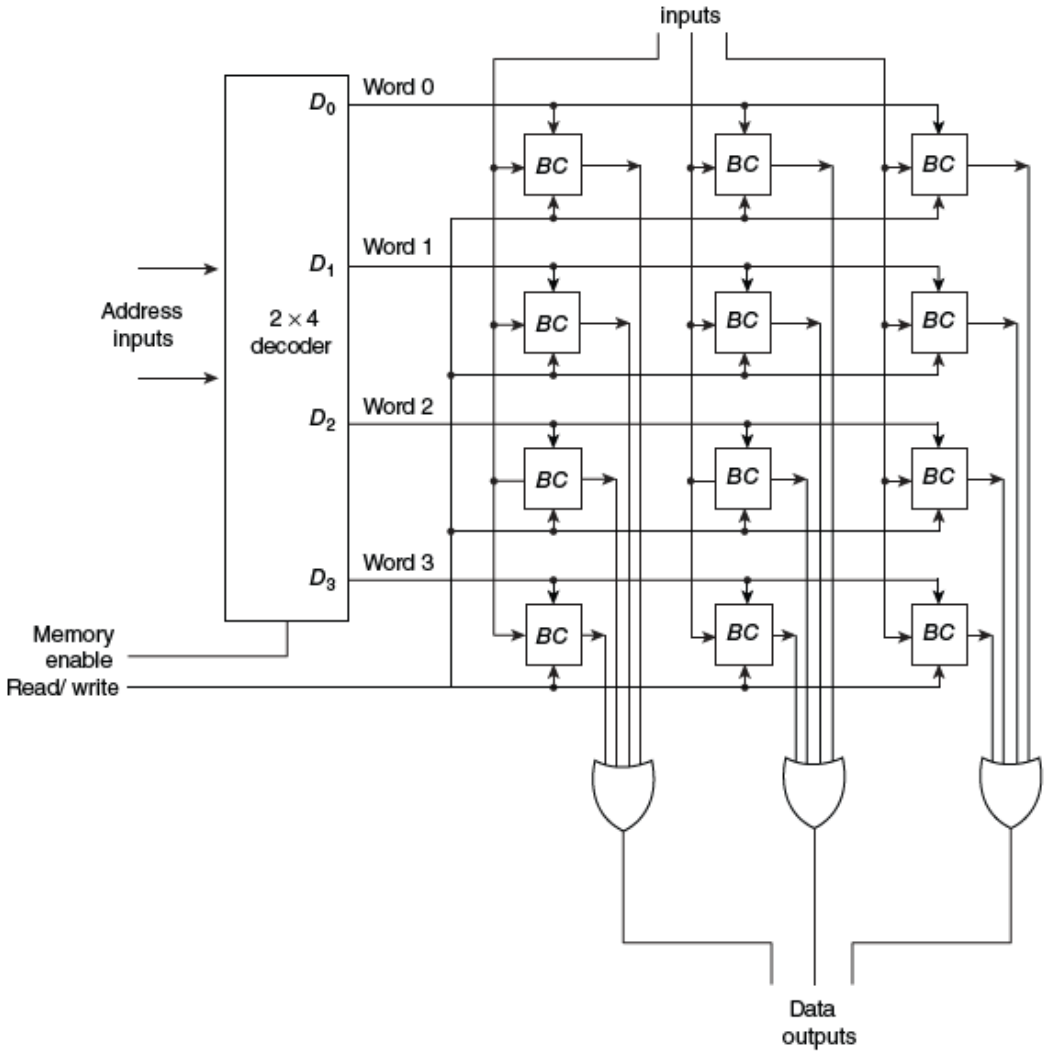
- **4 words** of **3-bit** each.
- BC: Binary cell
- 2 address i/ps: to **2-to4 line decoder**.
- **Decoder** enabled by:
  - Memory enable i/p.
    - 0: no memory word selected.
    - 1: one of the 4 selected (address lines).

Integrated Circuits memory unit (RAM)



- Read/write:
  - 1:
    - Bits of selected word go through **3 OR gates** to o/p.
    - **Non-selected cells** produce **0's** at the i/ps of OR.

Integrated Circuits memory unit (RAM)

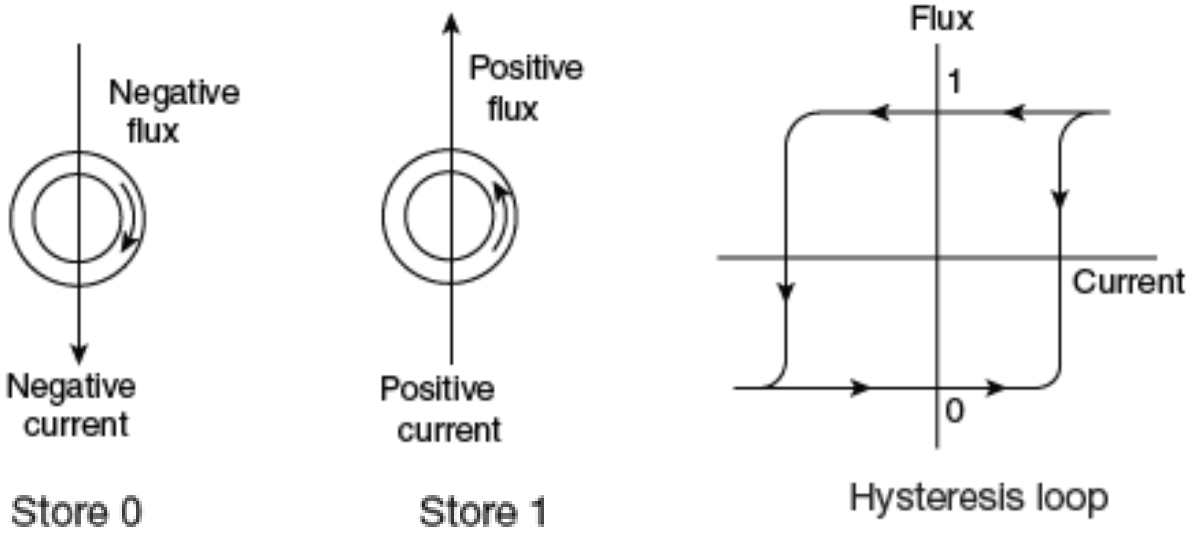


- Read/write:
  - 0:
    - Information in **i/p lines** transferred into BCs of selected word.
    - **Non-selected** BCs in other words **disabled** by their **selection i/ps** and their values are unchanged.
- With memory-enable=0
  - Contents of all cells in memory unchanged (regardless of read/write)

Magnetic core memory

- Magnetic core: Doughnut-shaped toroid.
- A **FF** requires **one physical quantity** for operation.
- Magnetic core: Uses **3** physical quantities.
  - Current, Magnetic flux and voltage
- **Excitation**: By **current pulse**
- Binary information represented by the **direction of magnetic flux** within core.
- O/p extracted as **voltage pulse**.

Magnetic core memory

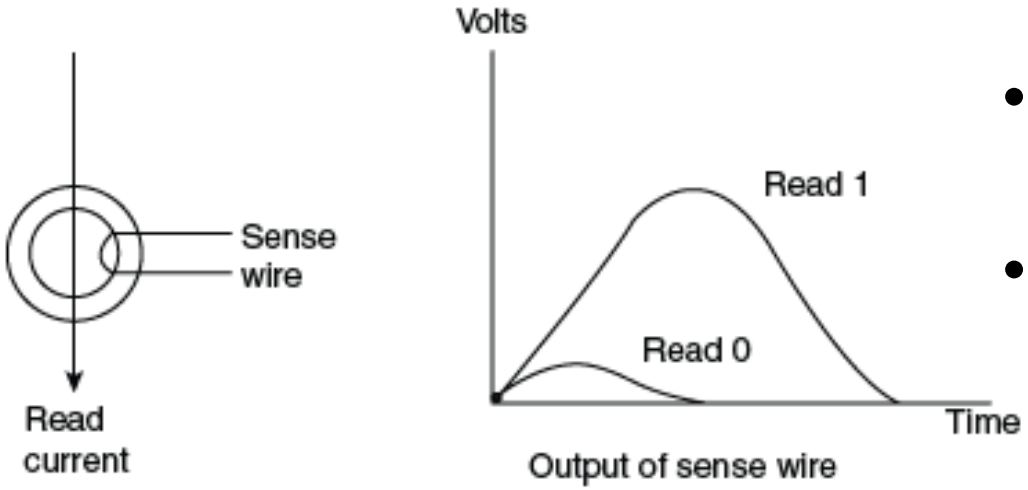


- **Hysteresis loop**, square loop: With **zero current**, a flux either
  - **positive** (counter clockwise)
  - **Negative** (clockwise)

remains in the magnetized core.

- **Current pulse** in the winding can **shift** the **direction** of magnetization.
- **Path flux follows** with **current pulse** indicated by arrows in **hysteresis** loop.

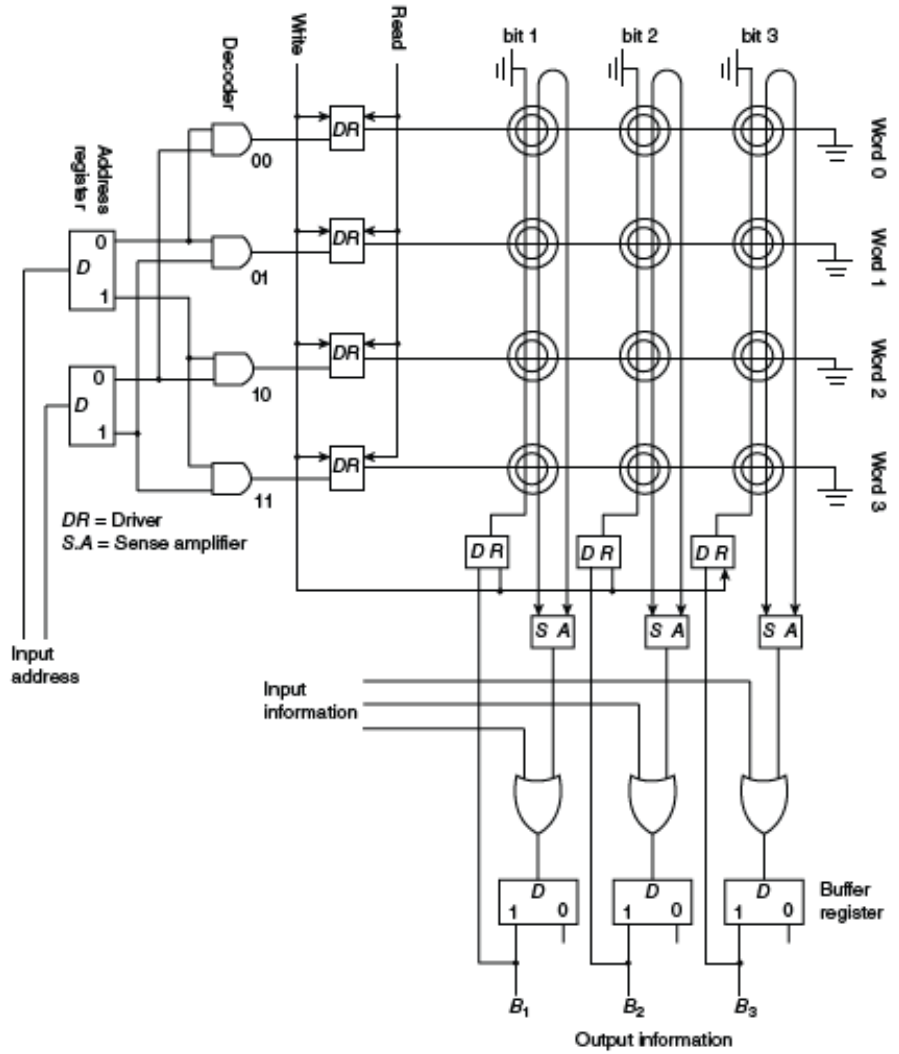
Magnetic core memory (Reading)



- **Reading**: Flux cannot be detected unless it is **changing**.
- Flux change wrt. time **induces** a **voltage** in wire that links the core.

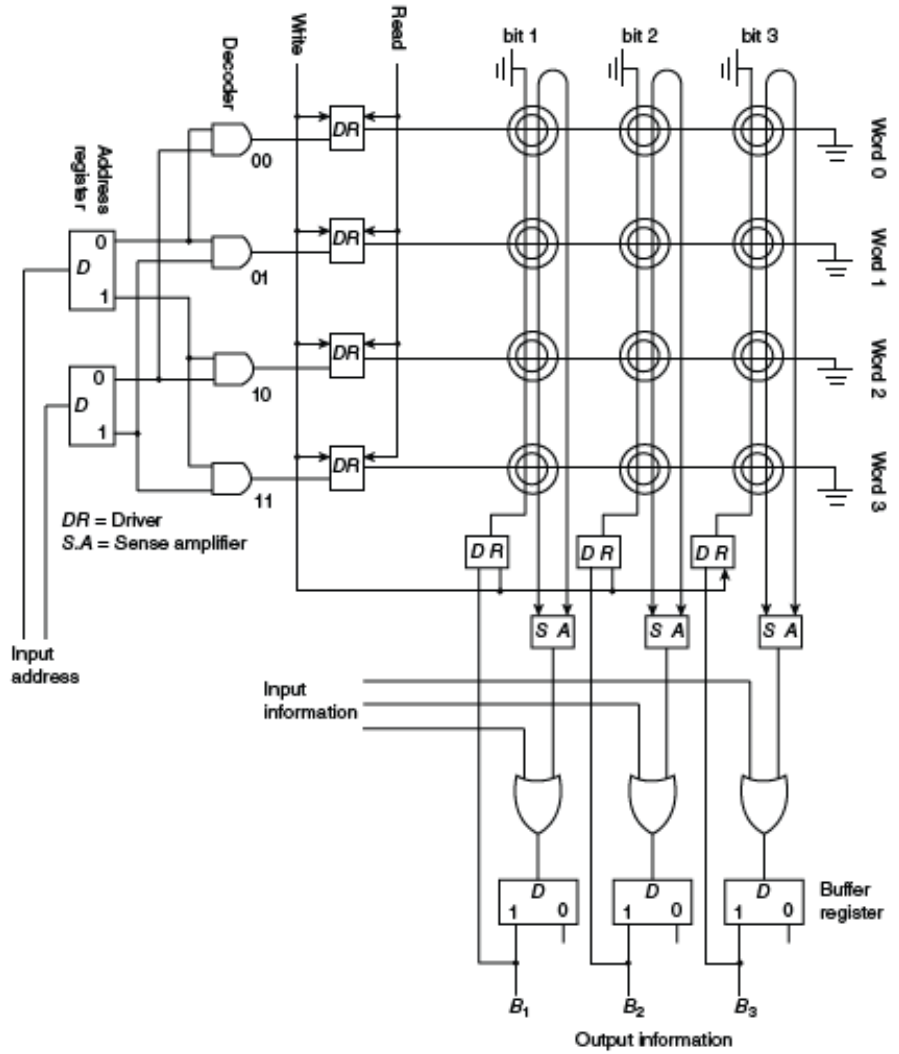
- Read out: A **current** in the **negative** direction is applied
  - 1 state:
    - Current **reverses** the **direction** of magnetization.
    - **Resulting flux change** produces a **voltage pulse** in sense wire.
  - 0 state:
    - Core **magnetization not changed**.
    - Very **slight disturbance** of magnetic flux. (**small** o/p voltage in sense wire.)

Magnetic core memory unit



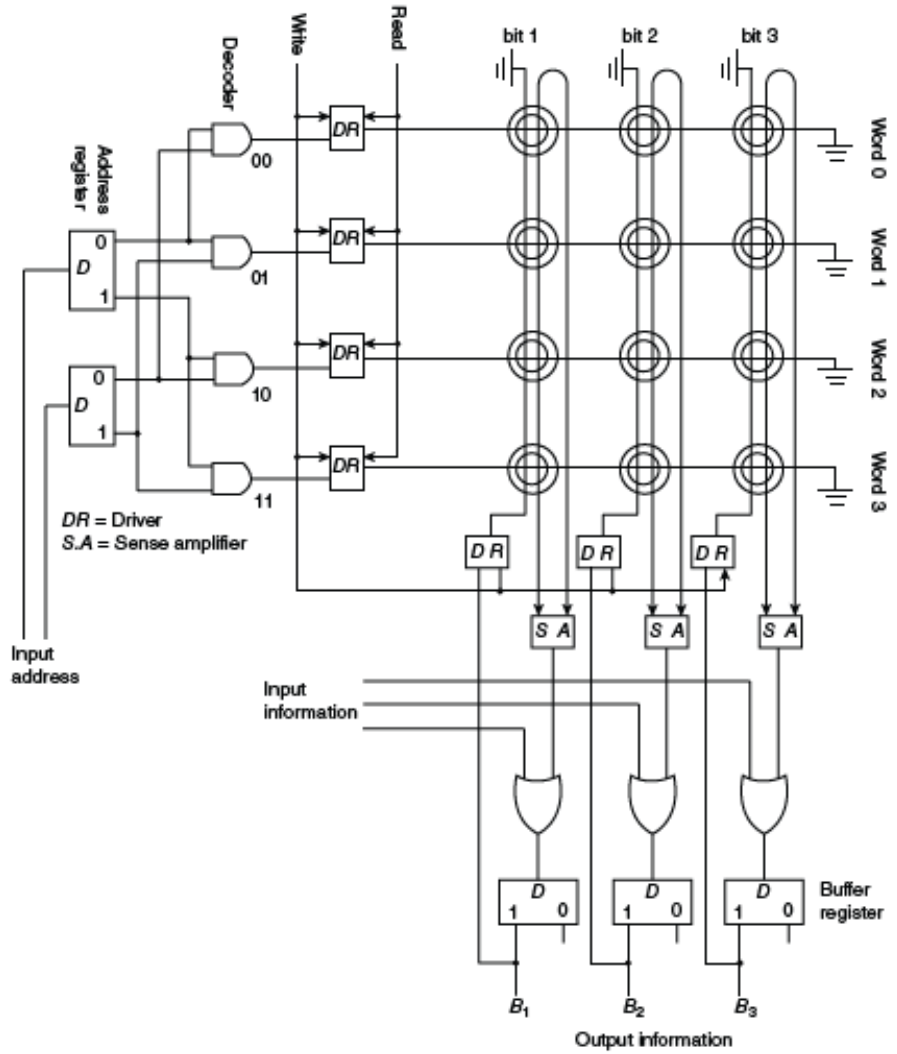
- **4 words** with **3-bits** each.
- **Binary cell**: **Magnetic core** with its wires.
- **Excitation** of core:
  - **Current** pulse generated in driver (**DR**)
- **O/p**: goes through a Sense Amplifier (**SA**)
  - **Set** corresponding **FFs** in buffer register.

Magnetic core memory unit



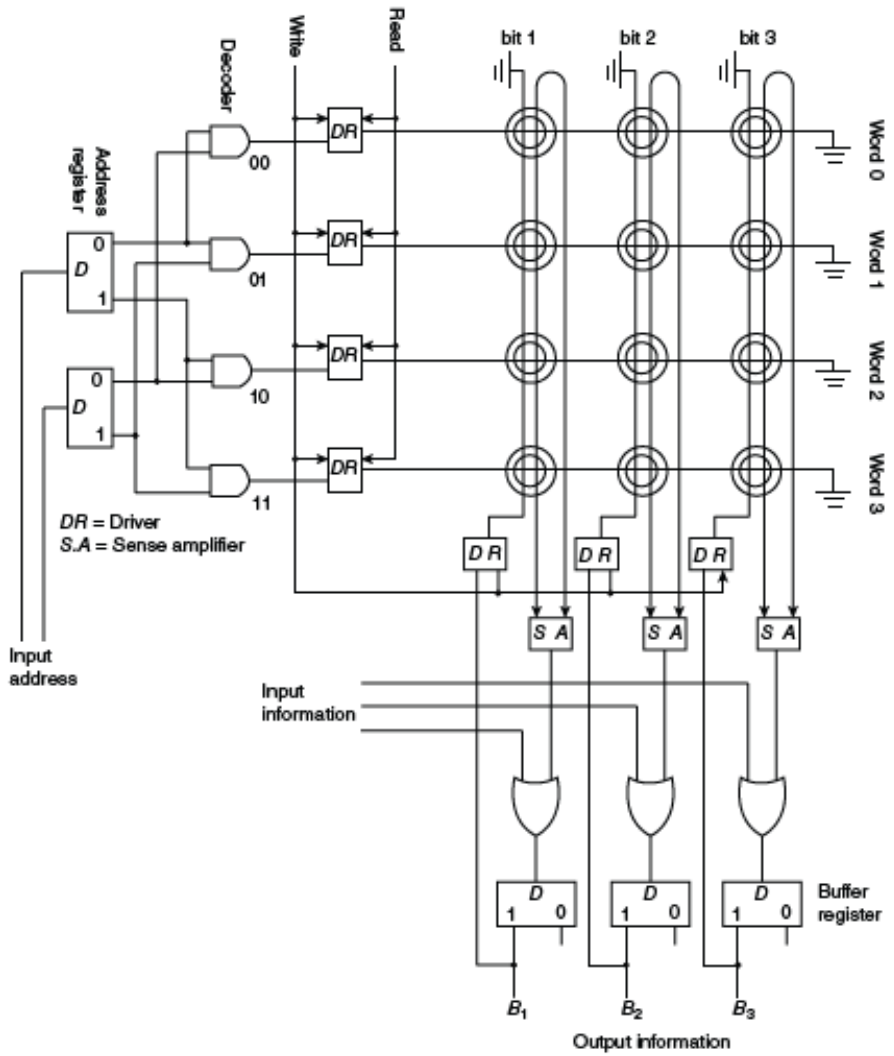
- **3 wires** link each core:
  - **Word wire**: Excited by word driver and goes through thr**ee cores of a word**.
  - **Bit wire**: Excited by bit driver and goes through **four cores** in **same bit position**.
  - **Sense wire**: links the same cores as the bit wires.
    - Applied to SA
- Sense Amplifier (SA):
  - **Shapes** voltage pulse when **1 is read**.
  - **Rejects** small disturbance when **0 is read**.

Magnetic core memory unit (Read)



- **Word driver** pulse applied to the cores of the words selected by decoder.
- **Read current**
  - is in **negative** direction.
  - **all cores** of selected core go to **‘0’** state.
- Cores with **1**
  - **Switch** their **flux**
  - **Induce** a **voltage** into their sense wire.
  - Voltage pulse is **amplified** in **SA**
  - **Sets** the corresponding **FF** in buffer register.
- Cores with **0**: **Flux** is **not changed**.

Magnetic core memory unit (Write)



- **Buffer register** holds the information to be stored in the word specified by address register.
- **Assume** that all cores in the selected word are **initially cleared**.
- For **writing 1**: A current pulse is generated **simultaneously** in
  - **Word driver** selected by decoder
  - **Bit driver** with corresponding buffer register **FF=1**.
- Both current in **positive direction**.
  - Magnitude is only **half needed** to switch the flux to 1 state.
- Direction of magnetization **does not change** if it receives **only one half** current.

Complete picture

- Read:
  - Selected word is **destroyed**.
  - Should be followed by **restoration**
  - Restore: **equivalent** to **write** operation. (Buffer register to selected word)
- Write:
  - Cores are **initially cleared**.
  - **Equivalent** to **Read** operation. (destruction of stored information)
  - But **prevents** the **Read information** from **reaching buffer register** by **inhibiting SA**.
- Restore and clear cycles **initiated** by memory **internal control** .
  - Outside world memory unit appears nondestructive.

Review

- Flip-Flops
- Triggering of Flip-Flops
- Analysis of clocked sequential circuits
- State reduction and Assignment
- Flip-Flop Excitation Tables
- Conversion of one FF to another
- Design Procedure
- Design of counters
- Shift registers