April 13, 2026

Q. Show the values of rax, rbx, and rcx after each instruction.

| | rax | rbx | rcx | notes |
|---|---|---|---|---|
| Initial state | 0 | 0 | 0 | |
| mov rax, 5 | 5 | 0 | 0 | |
| mov rbx, 4 | 5 | 4 | 0 | |
| add rbx, rax | 5 | 9 | 0 | rbx = 4+5 |
| shl rax, 1 | 10 | 9 | 0 | left shift by 1 = x2 |
| xor rbx, rax | 10 | 3 | 0 | |

Aside. Shifts.

★ To DO: mind your shifts.

shl

□□100 (4)

←

□1000 (8)

←

10000 (16)

Aside. XOR

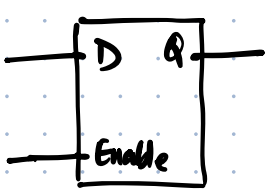
| | | |
|---|---|---|
| rax | 10 : | 1010 |
| rbx | 9 : | 1001 |
| XOR | 3 | 0011 |

Q. What is the difference between a D-flipflop and a D-latch?

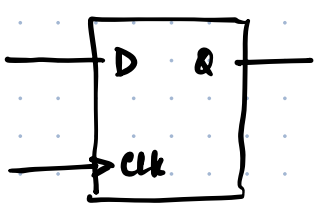
Aside

```
z = x & y;
z = x && y;
```

Bit-wise AND operator.

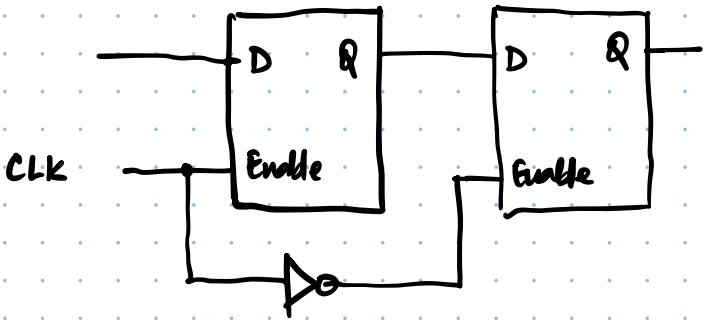


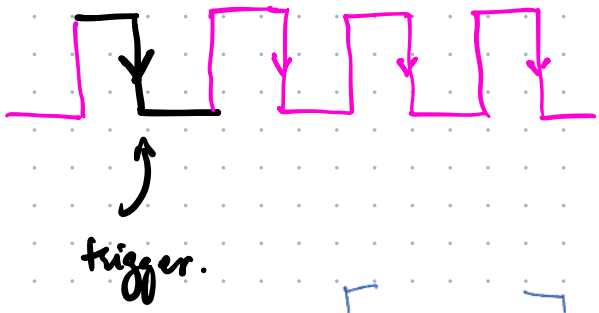

D latch




D flip flop

Q. Construct a D flip flop using one or more D latches?




* Mind your latches and flip flops.

Q. Cache. A processor uses direct-mapped cache with:

(i) Cache capacity: 32 blocks (5 bits)

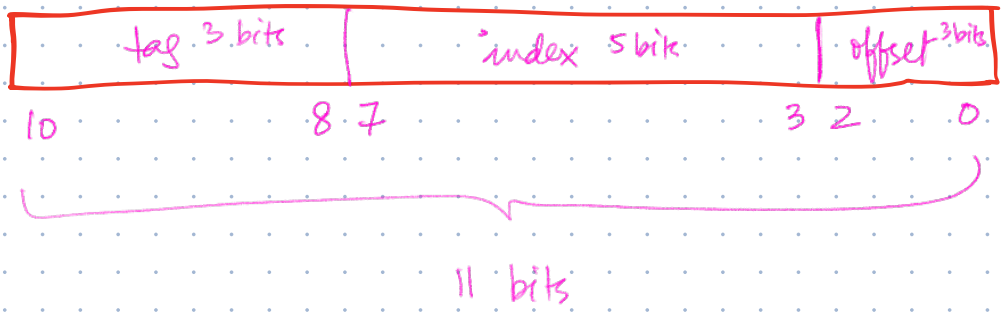
(ii) Block size: 8 bytes (3 bits)

(iii) Main memory size: 2048 bytes (11 bits)

$\lceil \log_2 2048 \rceil$

$2^n$

1. How many bits are used for the offset, index and tag? Identify which bit position each field occupies?




2. Compute Average Memory Access time (AMAT)?

(i) Cache hit time: 1 ns

(ii) Memory access time: 50 ns

(iii) Cache miss rate: 5 %

time units ↑↑

AMAT = $0.95 \times 1 + 0.05 \times 50$

= ns

Q. A direct-mapped cache has 4 slots (indexed 0–3). A block at address n maps to slot n mod 4. ✓

Trace the access sequence below, marking each access as HIT or MISS.
Show the cache state
after each access.
Access sequence: 0, 1, 2, 3, 0, 4, 0, 3, 1, 2

| Access | Slot | Cache [0,1,2,3] | Result |
|---|---|---|---|
| 0 | 0 | [0, -, -, -] | MISS |
| 1 | 1 | [0, 1, -, -] | MISS |
| 2 | 2 | [0, 1, 2, -] | MISS |
| 3 | 3 | [0, 1, 2, 3] | MISS |
| 0 | 0 | [0, 1, 2, 3] | HIT |
| 4 | 4 | [4, 1, 2, 3] | MISS |
| 0 | 0 | [0, 1, 2, 3] | MISS |
| 3 | 3 | [0, 1, 2, 3] | HIT |
| 1 | 1 | [0, 1, 2, 3] | HIT |
| 2 | 2 | [0, 1, 2, 3] | HIT |

Hit rate = $\frac{4}{10}$

= 40%

Q. Virtual Memory

A system uses paging with the following parameters:
- Virtual address space: 16 bits (64 KB)
- Page size: 256 bytes
- Physical memory: 4096 bytes (4 KB)

(i) How many bits are used for the page offset?
(ii) How many bits for the Virtual Page Number (VPN)?
(iii) How many virtual pages exist?
(iv) How many physical frames exist?

Offset: $\lceil \log_2(\text{page size}) \rceil = \lceil \log_2(256) \rceil = 8$ bits

# of physical pages = frames = $\frac{\text{memory size}}{\text{page size}} = \frac{4096}{256} = 16$

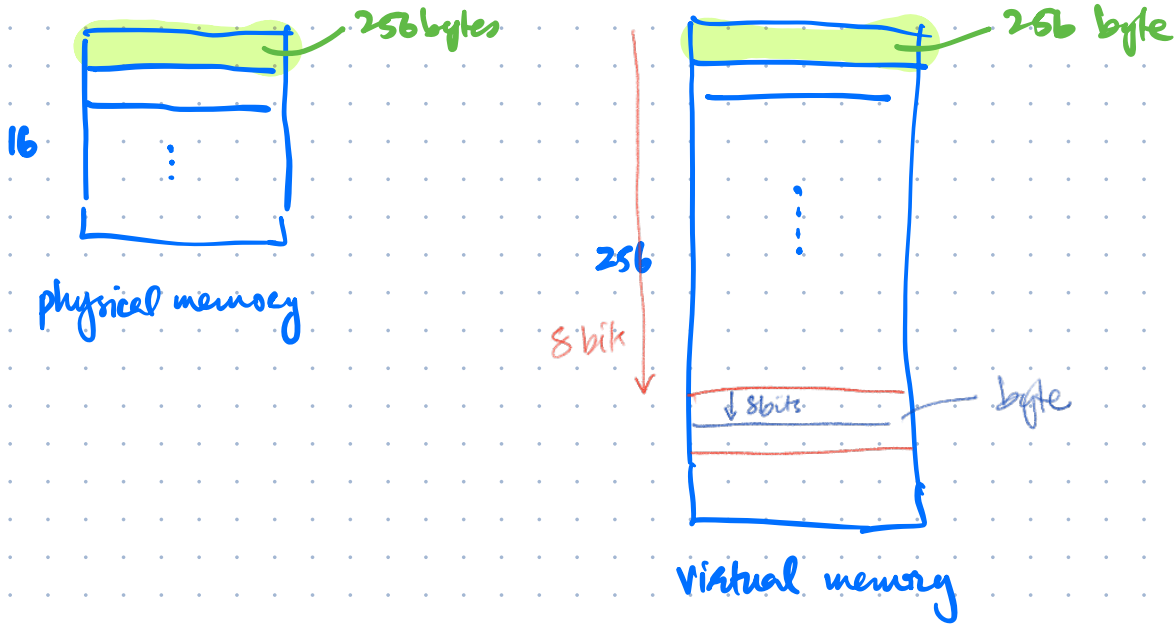
total address space = 16 bits

VPN = total address bits − offset bits

= 16 − 8

= 8 bits

Distinct VPN = $2^8$ = 256

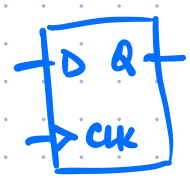



Q. Construct a modulo-5 synchronous counter using D flip flops.

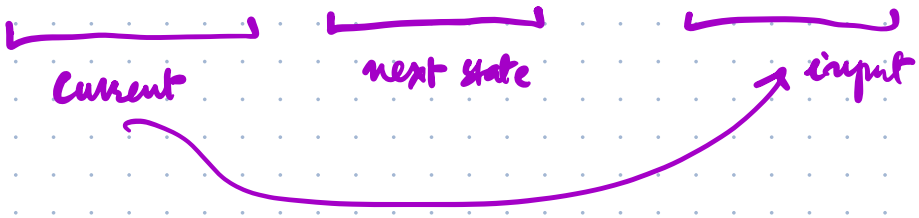
0 → 1 → 2 → 3 → 4 → 0 → 1 → ...

000 001 010 011 100 000 001

Inside:




| $Q_2$ | $Q_1$ | $Q_0$ | next state $Q_2'$ | $Q_1'$ | $Q_0'$ | $D_2$ | $D_1$ | $D_0$ |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | x | x | x | x | x | x |
| 1 | 1 | 0 | x | x | x | x | x | x |
| 1 | 1 | 1 | x | x | x | x | x | x |




Let's look at $D_2$

| $Q_2 Q_1$ \ $Q_0$ | 0 | 1 |
|---|---|---|
| 00 | | |
| 01 | | 1 |
| 11 | X | X |
| 10 | | X |

$D_2 = Q_1 \cdot Q_0$

$D_1 = \bar{Q}_1 Q_0 + Q_1 \bar{Q}_0$

Similarly $D_1 = Q_1 \oplus Q_0$ (XOR)

$D_0 = \bar{Q}_2 \cdot \bar{Q}_0$