

# 4x4 RAM (4 words x 4 bits)

- Synchronous write, Asynchronous Read
- A small memory with  $4 \text{ locat}^n$  (address 2bits) each 4 bits wide.

Write :- synchronous on posedge clk  
when  $en = 1$  and  $R\_w = 1$

→ \* write data in into reg [address]

Read :- combinational / Asynchronous  
when  $en = 1$  and  $R\_w = 0$

→ \* drives data out from reg [address]

\* Data-out is driven high impedance (1'bz)  
during write cycle. Memory initialized to 0

## Code Explanation

① Declaration :- reg [0..3] is 4x4 storage

• data\_out is a reg [3:0] o/p

• address is 2bits

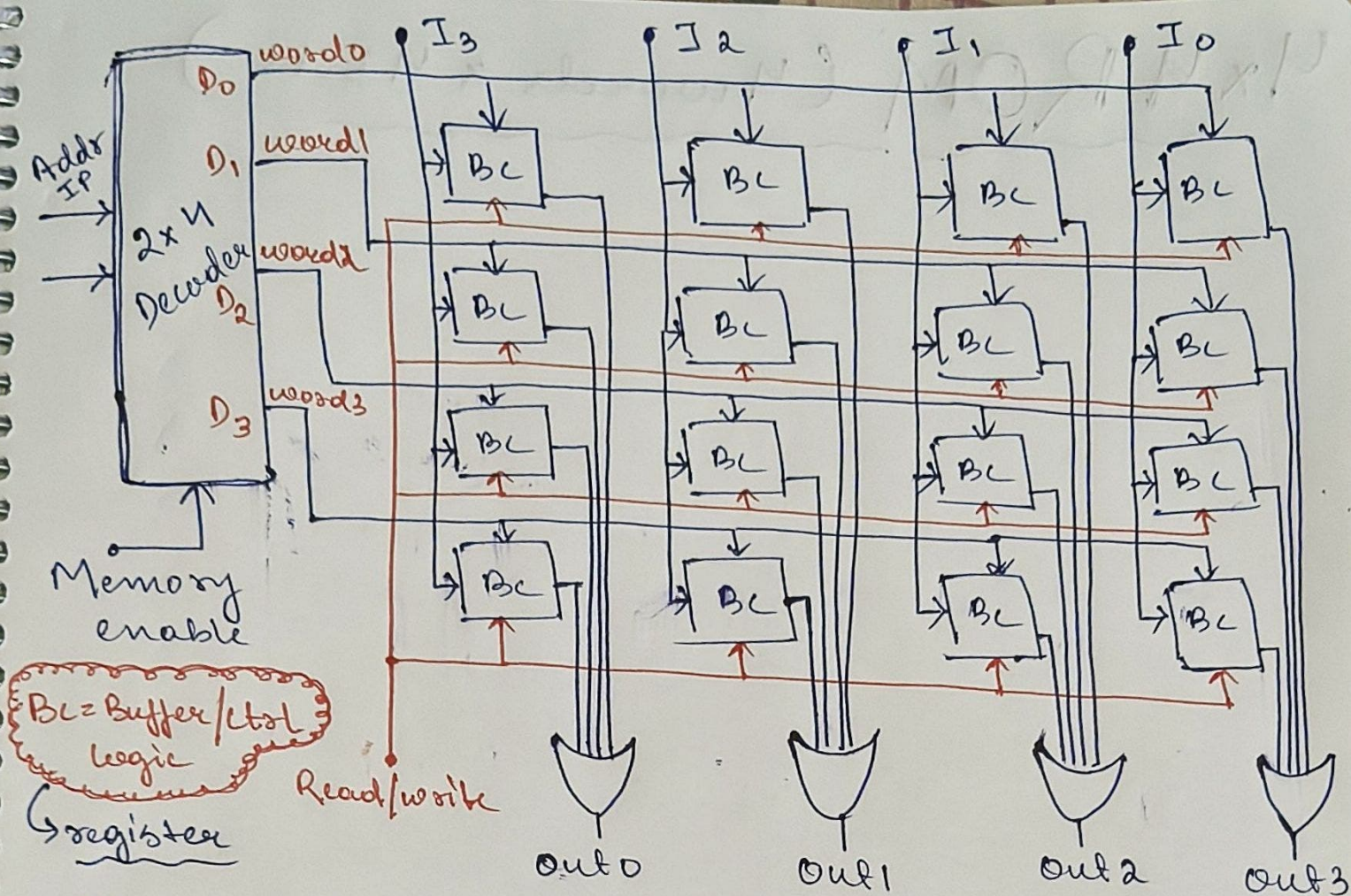
•  $R\_w = 1$ , write  $R\_w = 0$ , Read  
en enable memory

② Initial block :- data\_out = 0 and clears all  
reg to 0 at simulat<sup>n</sup> start

• Formula :- Memory capacity =  $2^n$  words x m bits/word

$n = \text{addr lines}$      $m = \text{no. of data lines (i/p / o/p)}$





### • Always block (write)

```

always @(posedge clk) begin
  if (en) begin
    if (R_w) begin
      register[address] = data_in;
      data_out = 4'b z;
    end
  end
end

```

on rising clk edge  
 en = 1 and R\_w = 1  
 data-in stored in  
 Register[Address] &  
 data-out = 1'b z  
 during write op<sup>n</sup>

### • Combinational block (Read)

```

always @(*) begin
  if (en) begin
    if (!R_w) data_out = register[address];
    else data_out = 4'b z;
  end
end

```

en = 1 and R\_w = 0  
 data-out immediately  
 reflect register[Address]

otherwise data out  
 is z (high impedance)