

4x4 RAM (4 words x 4 bits)

- Synchronous write, Asynchronous Read
- A small memory with 4 locat^n (address 2 bits) 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

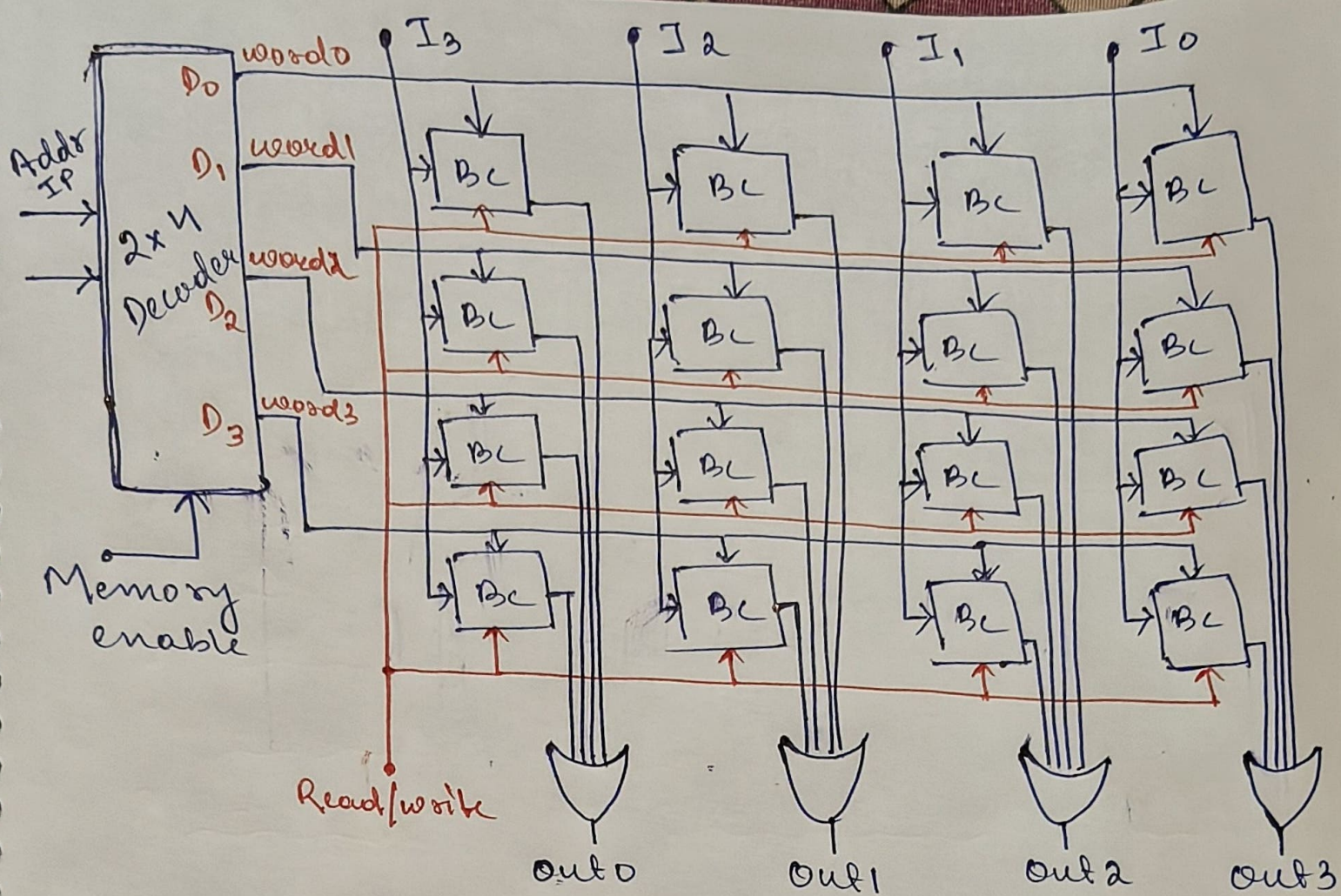
① Declaration :- reg [0..3] is 4x4 storage

• data_out is a reg [3:0] o/p

• address is 2 bits

• $R_w = 1$, write $R_w = 0$, Read
en enable memory

② Initial block :- data_out = 0 and clears all
reg to 0 at simulatⁿ start



• 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 opⁿ

• 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)