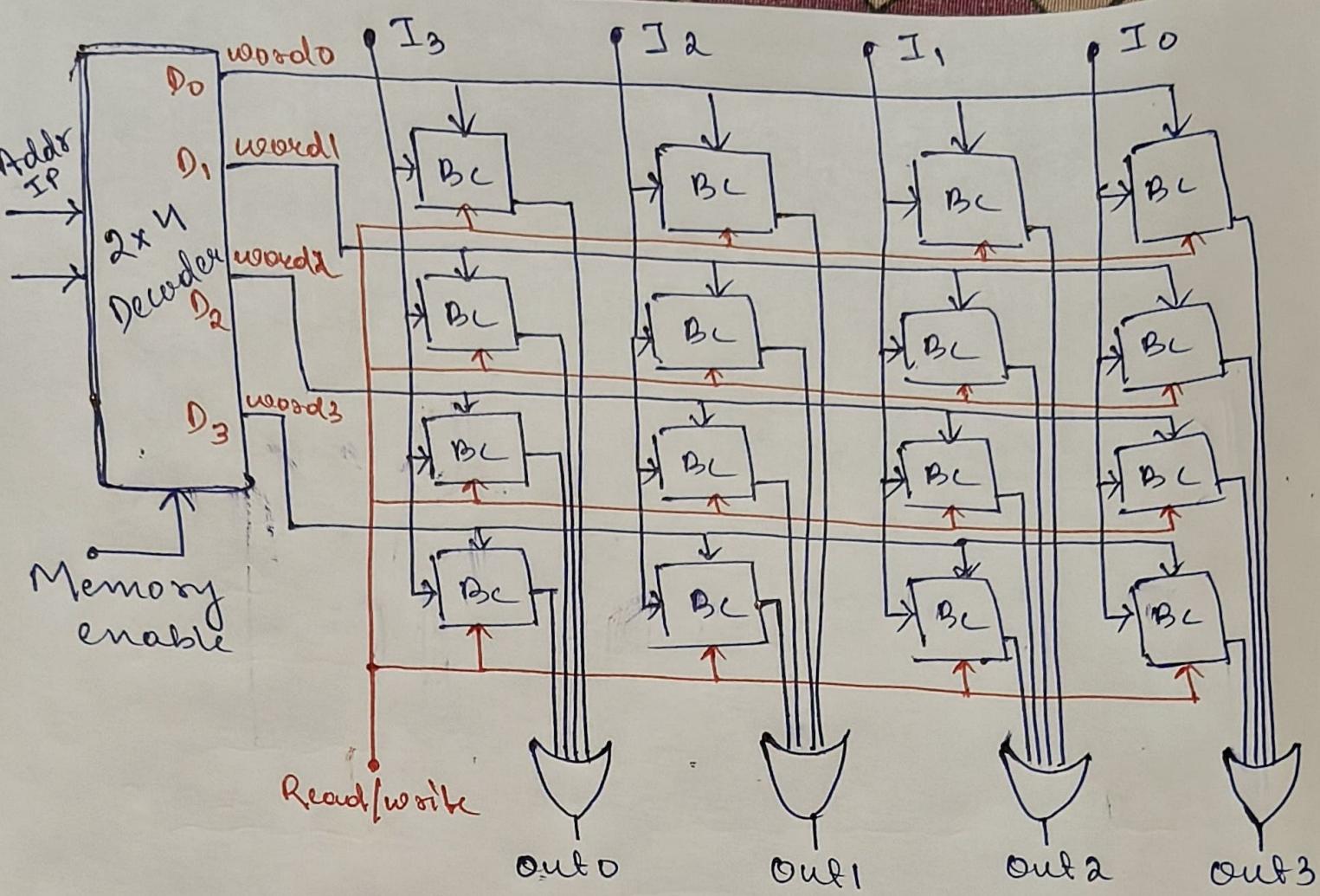


4x4 RAM (4 words x 4 bits)

- Synchronous write, Asynchronous Read
- A small memory with 4 location 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] of
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 simulation start



Always block (write)

```

always @ (posedge clk) begin
  if (en) begin
    if (!R_W) begin
      register[address] = data_in;
      data_out = 4'bzz;
    end
  end
end

```

} on rising CLK edge
 $en=1$ and $R_W=1$
 data-in stored in
 Register [Address] &
 $data_out = 4'bzz$
 during write op"

Combinational block (Read)

```

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

```

} $en=1$ and $R_W=0$
 $data_out$ immediately
 reflect register [Address]

} otherwise $data_out$
 is Z (high impedance)