4 x 4 RAM (4 words x 4 bits)

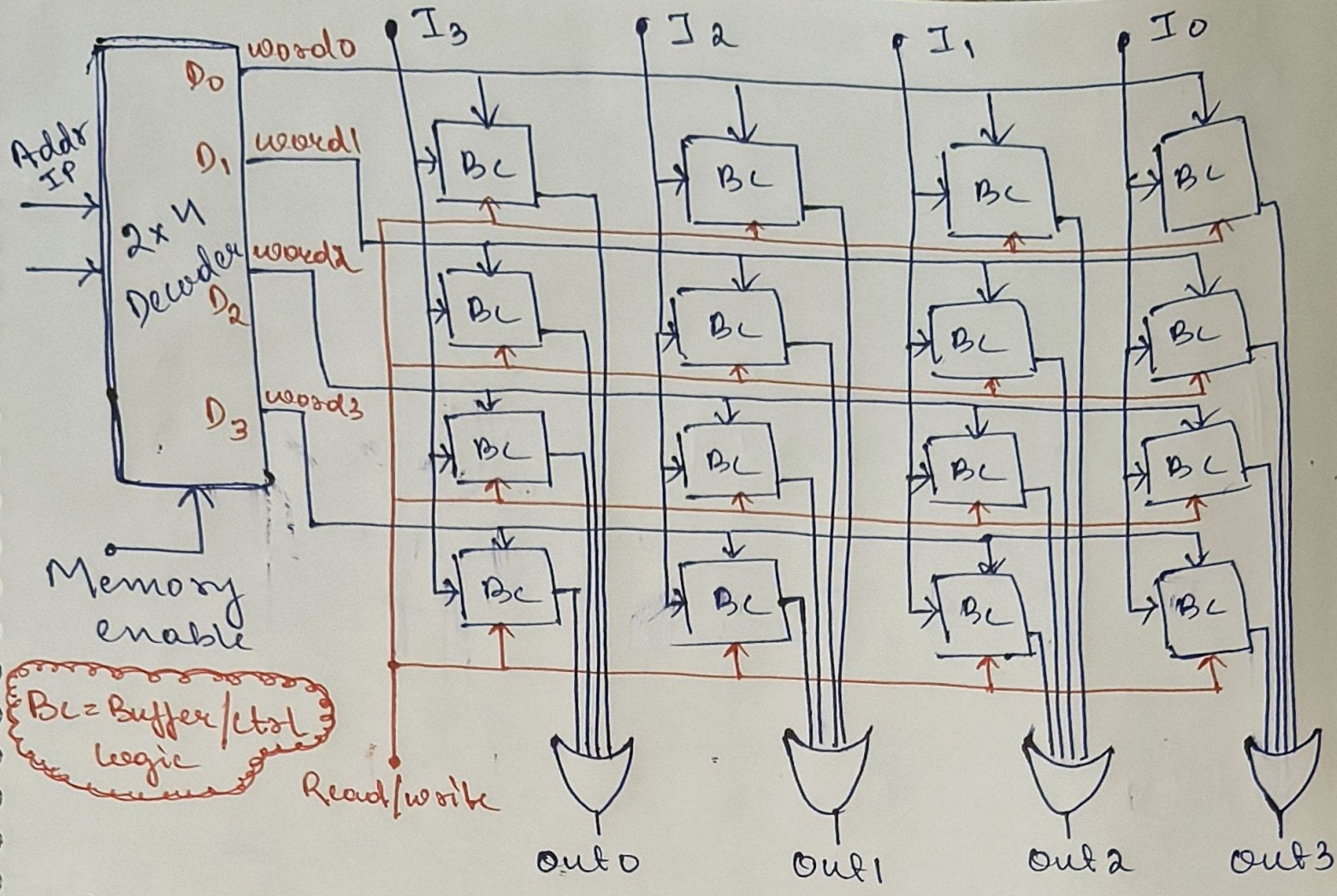
- Synchronous write, Asynchronous Read
- A small memory with 4 locations (address 2 bits) each 4 bits wide.
- Write :- synchronous on posedge clk
  - when en=1 and R_W=1
  - * write data_in into reg[address]
- Read :- combinational / Asynchronous
  - when en=1 and R_W=0
  - * drives data_out from reg[address]

* Data_out is driven high impedance (1'bz) during write cycle. Memory initialized to 0

- Code Explaination

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 simulation start



- Always block (write)

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

on rising clk edge en=1 and R_W=1 data_in stored in Register[Address] & data_out = 1'bz during write op^n

- Combinational block (Read)

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

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

otherwise data_out is z (high impedance)