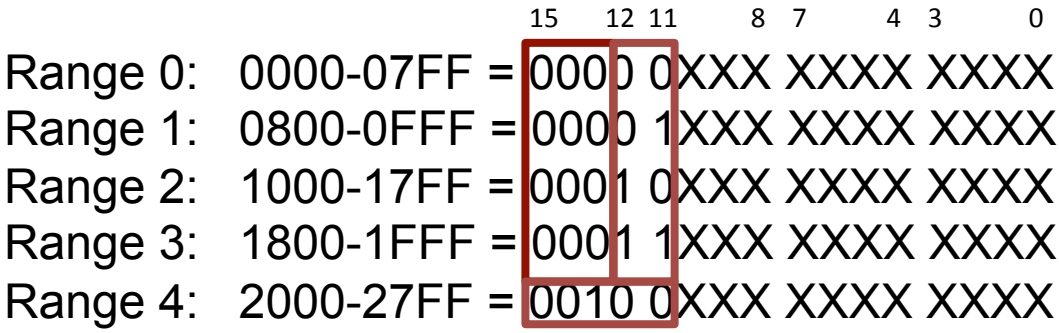
ECE 545

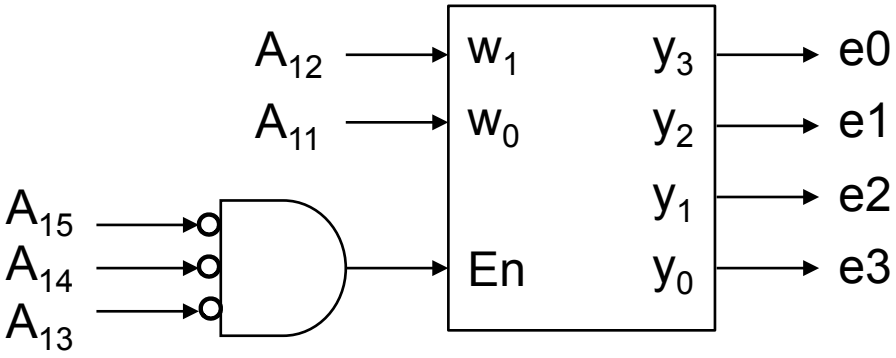
Midterm Exam

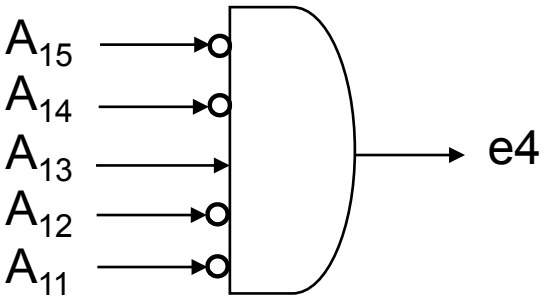
Fall 2013

Solutions

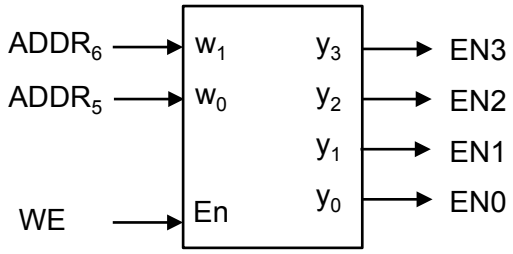
Problem 1

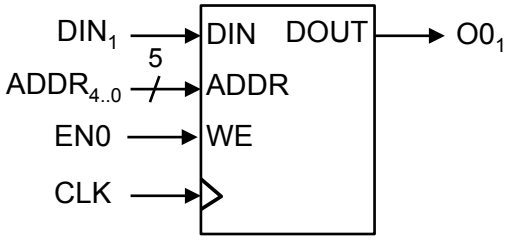

15 12 11 8 7 4 3 0
Range 0: 0000-07FF = 0000 0XXX XXXX XXXX
Range 1: 0800-0FFF = 0000 1XXX XXXX XXXX
Range 2: 1000-17FF = 0001 0XXX XXXX XXXX
Range 3: 1800-1FFF = 0001 1XXX XXXX XXXX
Range 4: 2000-27FF = 0010 0XXX XXXX XXXX

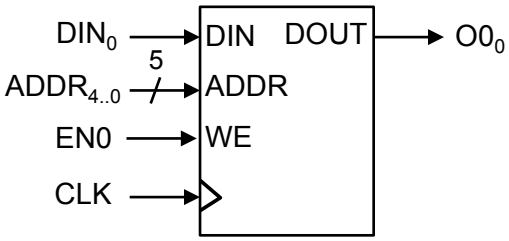

A12
A11
A15
A14
A13
w1
w0
En
y3
y2
y1
y0
e0
e1
e2
e3

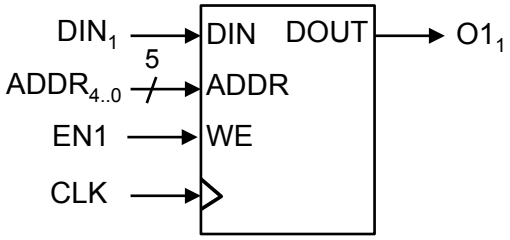

A15
A14
A13
A12
A11
e4

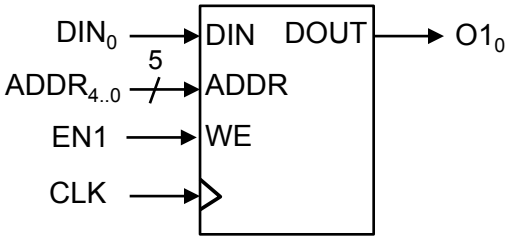
Problem 2

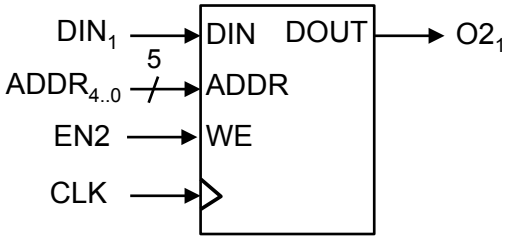

ADDR6
ADDR5
WE
w1
w0
En
y3
y2
y1
y0
EN3
EN2
EN1
EN0

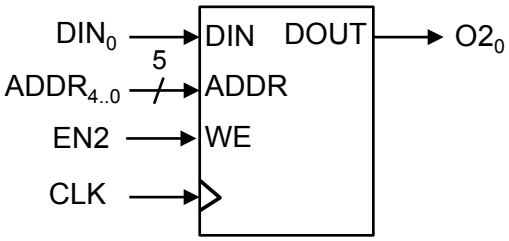

DIN1
5
ADDR4..0
EN0
CLK
DIN
DOUT
ADDR
WE
O01

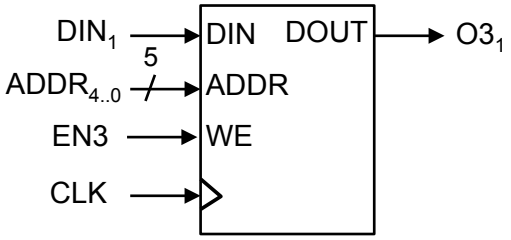

DIN0
5
ADDR4..0
EN0
CLK
DIN
DOUT
ADDR
WE
O00

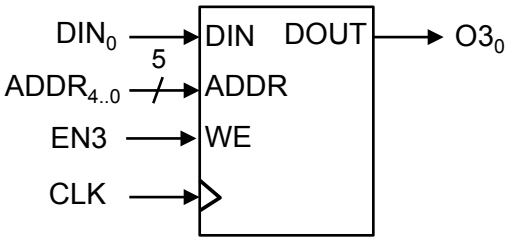

DIN1
5
ADDR4..0
EN1
CLK
DIN
DOUT
ADDR
WE
O11

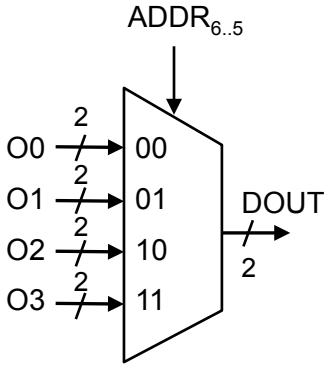

DIN0
5
ADDR4..0
EN1
CLK
DIN
DOUT
ADDR
WE
O10


DIN1
5
ADDR4..0
EN2
CLK
DIN
DOUT
ADDR
WE
O21

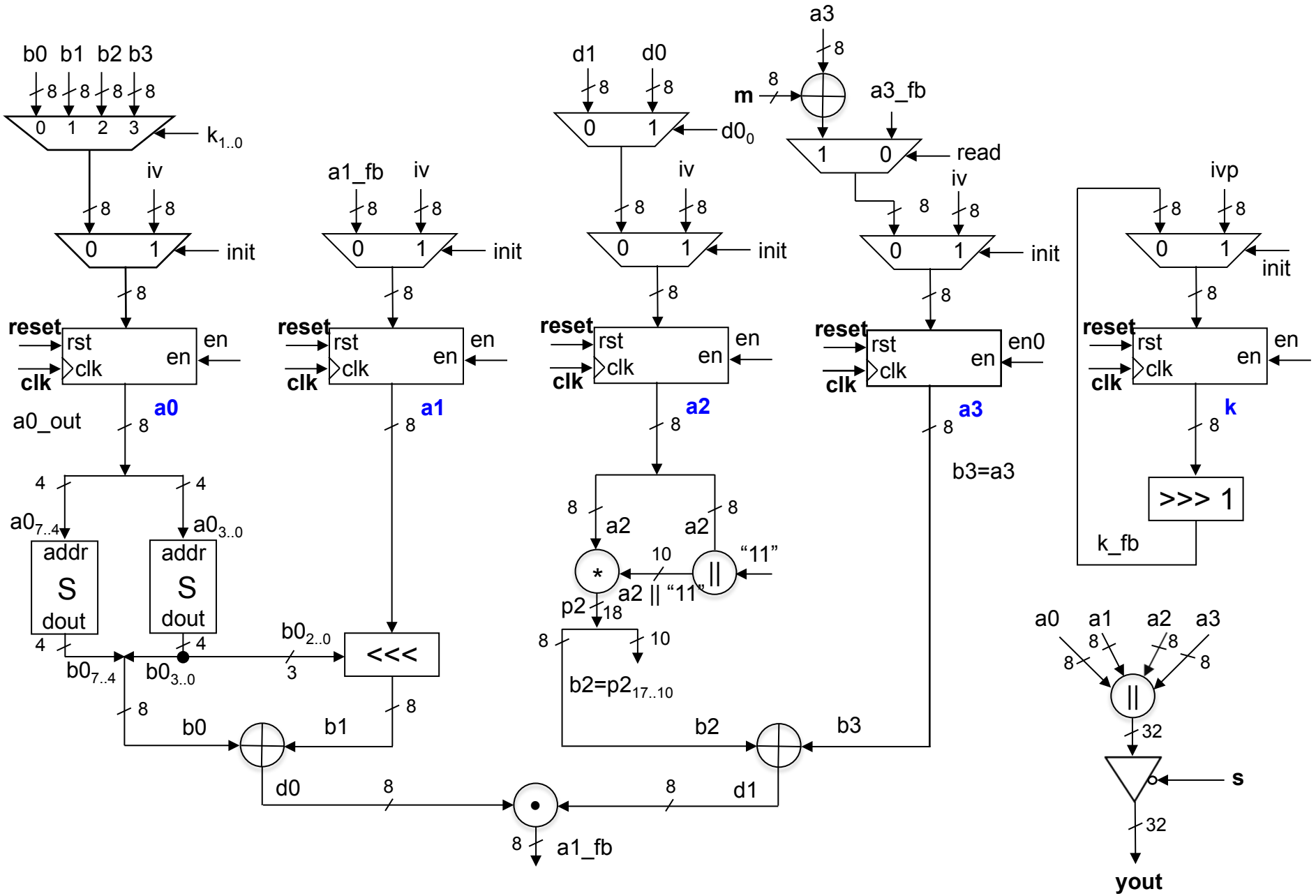

DIN0
5
ADDR4..0
EN2
CLK
DIN
DOUT
ADDR
WE
O20

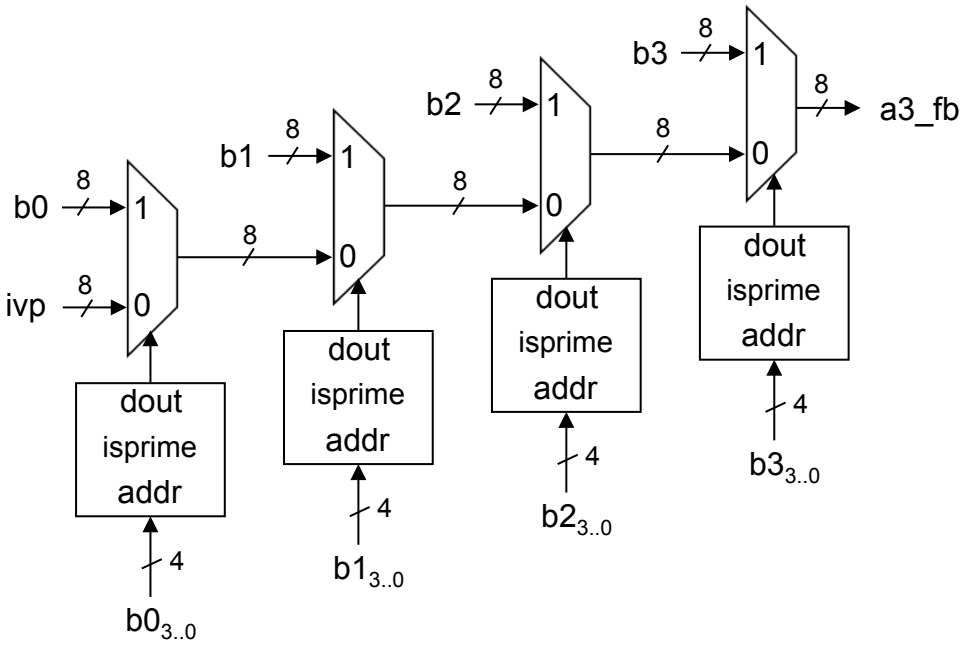

DIN1
5
ADDR4..0
EN3
CLK
DIN
DOUT
ADDR
WE
O31


DIN0
5
ADDR4..0
EN3
CLK
DIN
DOUT
ADDR
WE
O30


ADDR6..5
O0
O1
O2
O3
2
00
01
10
11
DOUT
2

Problem 3 – Task 1

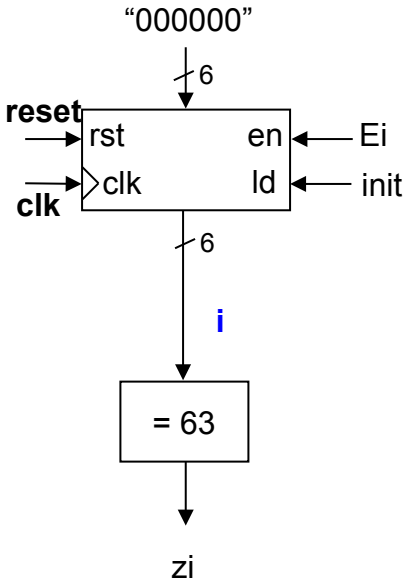


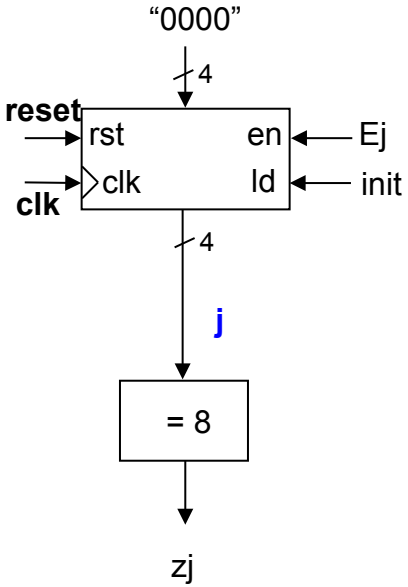


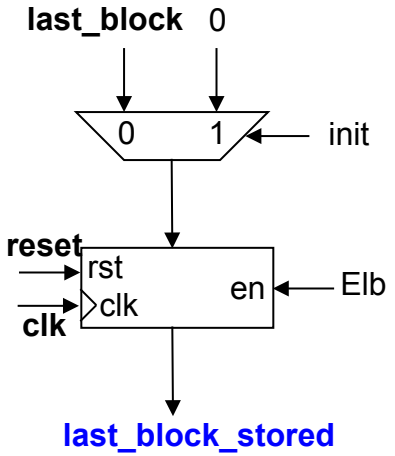
Optimizations:

b3 = (8*a3 ⊕ 6) >> 3 =
= (a3 || “110”) >> 3 = a3

$k = 0 \,||\, k_{7..1}$ if $k_0 = 0$
$k = 1 \,||\, k_{7..1}$ if $k_0 = 1$
thus,
$k = k_0 \,||\, k_{7..1} = k >>> 1$

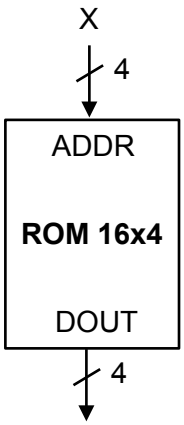






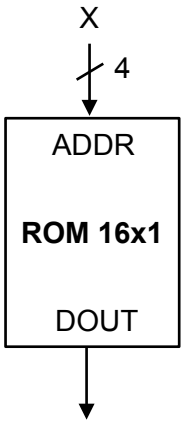
Problem 3 – Task 2

a) Substitution Y=S[X]



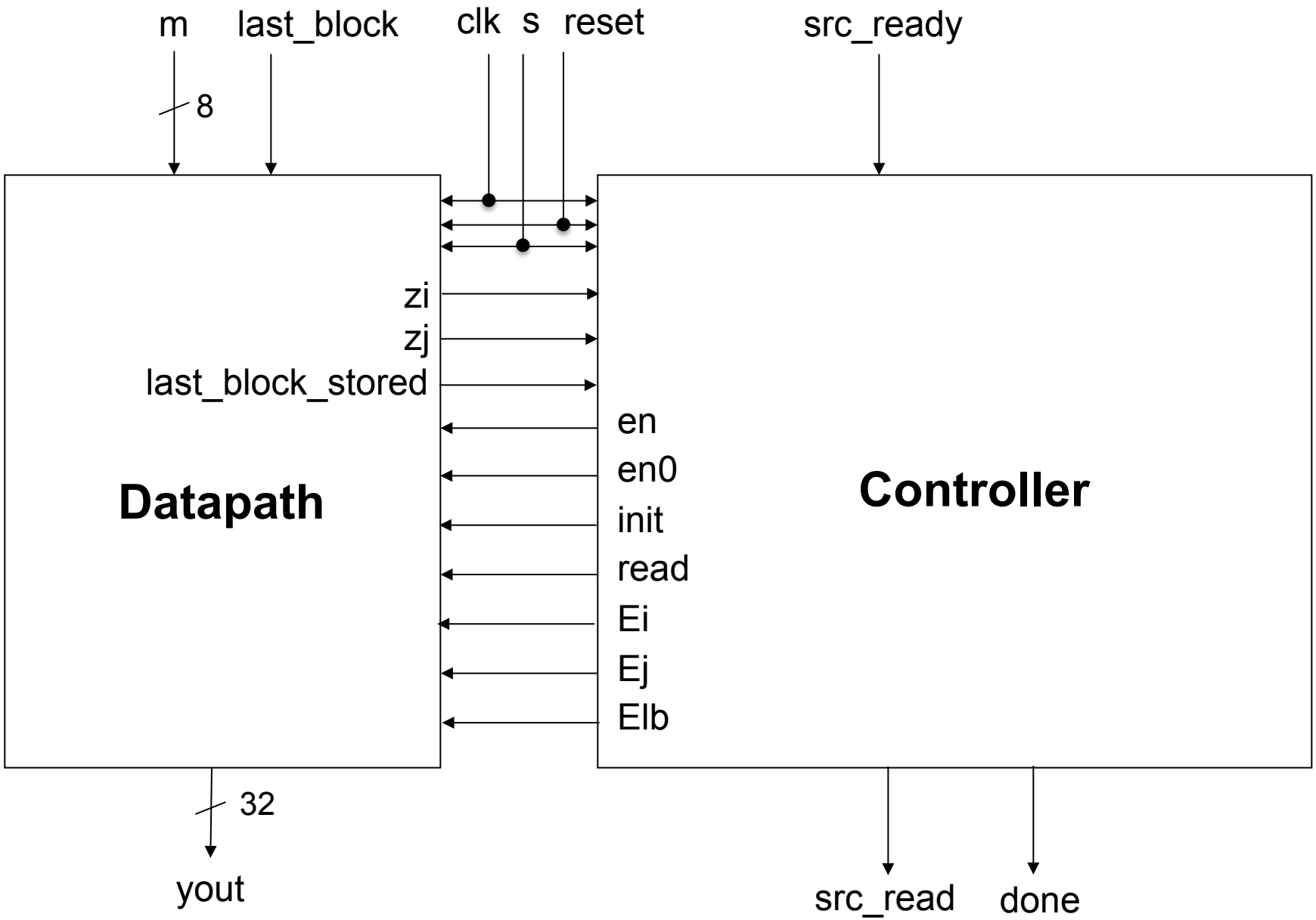
0	E
1	D
2	B
3	0
4	2
5	1
6	4
7	F
8	7
9	A
10	8
11	5
12	9
13	C
14	3
15	6

b) Function Y=isprime[X]



0	0
1	0
2	1
3	1
4	0
5	1
6	0
7	1
8	0
9	0
10	0
11	1
12	0
13	1
14	0
15	0

Problem 3 – Task 3



Problem 4

```
library ieee;
use ieee.std_logic_1164.all;

entity misr is
        generic (C : std_logic_vector(7 downto 0) );
    port (
        -- inputs
        clk     :       in      std_logic;
        rst     :       in      std_logic;
        en      :       in      std_logic;
        D       :       in      std_logic_vector (7 downto 0);

        -- outputs
        Q_out:          out     std_logic_vector (7 downto 0)
       );
end misr;

architecture mixed of misr is

-- intermediate signals
  signal  Q : std_logic_vector (7 downto 0);
  signal  Q0_replicated : std_logic_vector (7 downto 0);
  signal  d_ff_in : std_logic_vector (7 downto 0);

begin

        Q0_replicated <= (others => Q(0));
        d_ff_in <= D xor ('0' & Q(7 downto 1)) xor (C and Q0_replicated);


-- D flip flop operation
        D_FFs: process (rst, clk)
        begin
                if (rst = '1') then
                        Q <= (others => '0');
                elsif rising_edge(clk) then
                        if(en = '1') then
                                Q <= d_ff_in;
                        end if;
                end if;
        end process;

        Q_out <= Q;
end mixed;
```