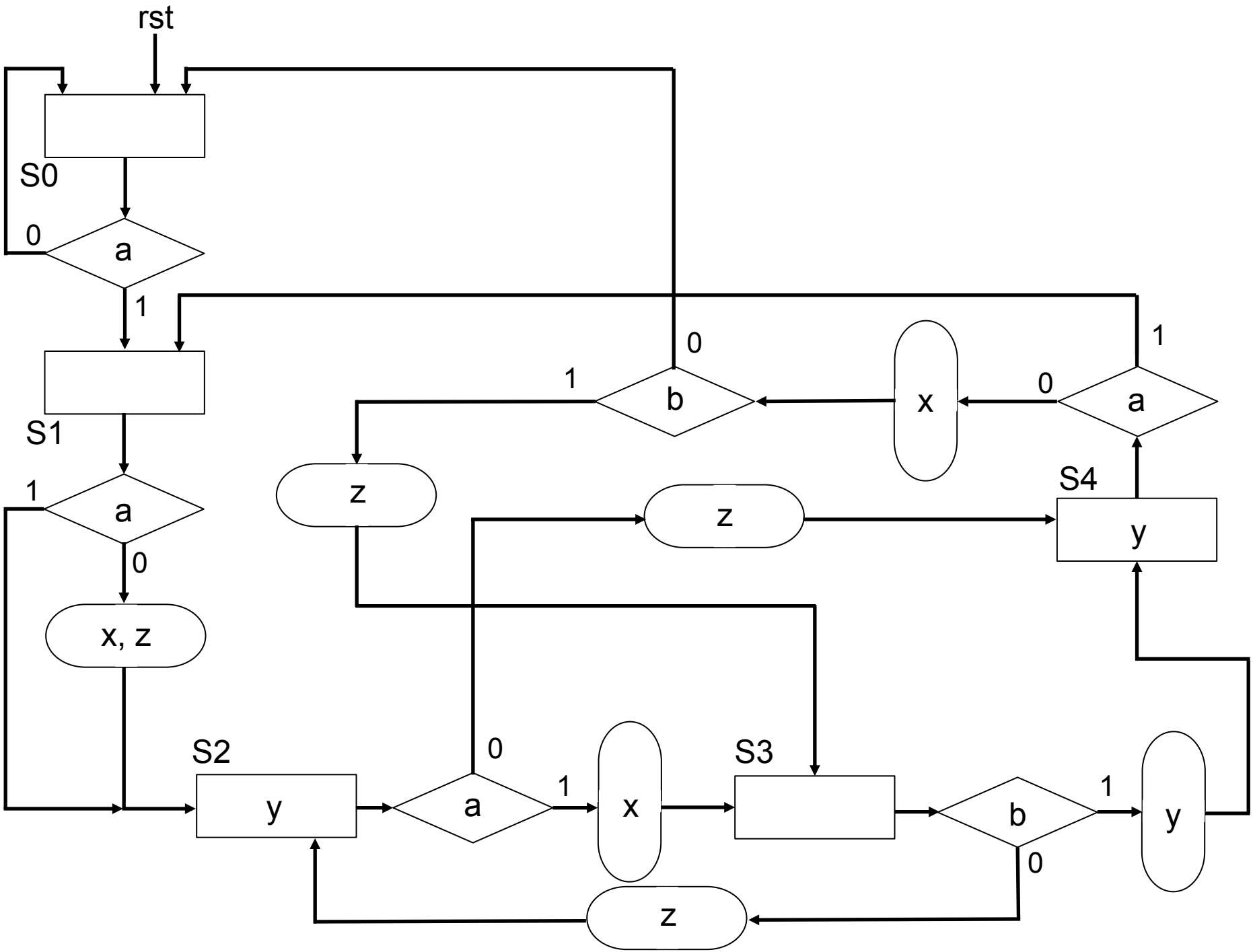
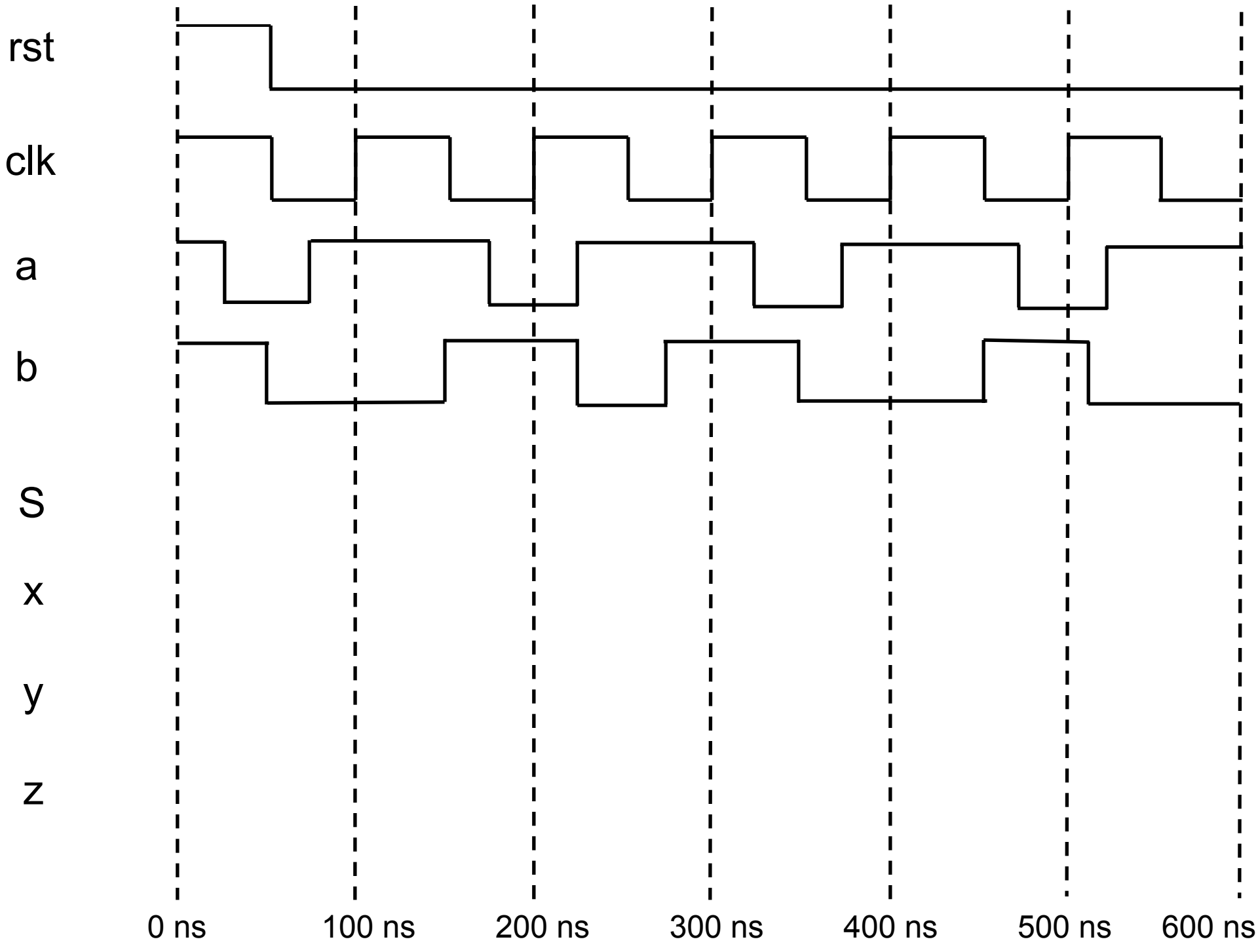
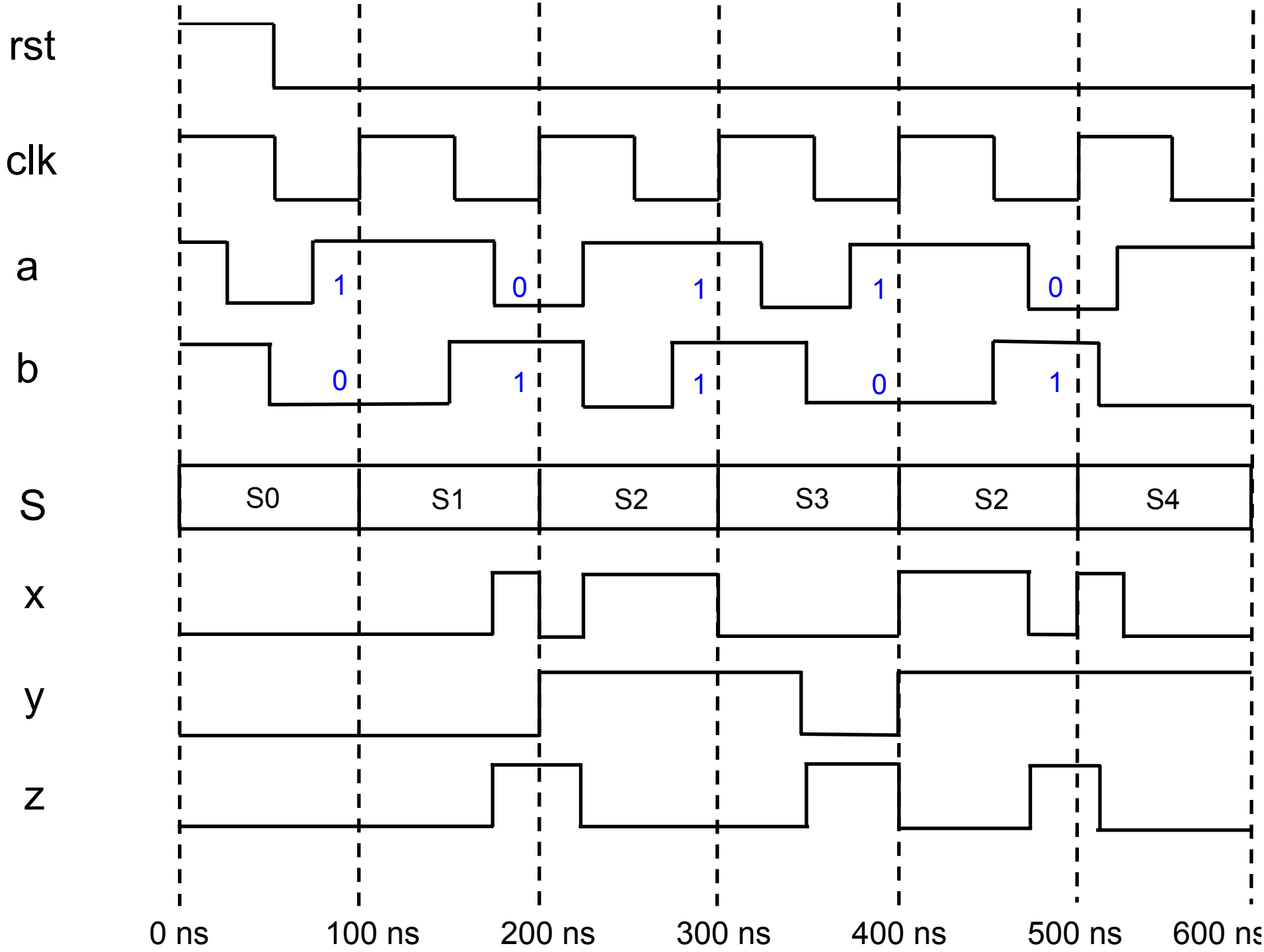
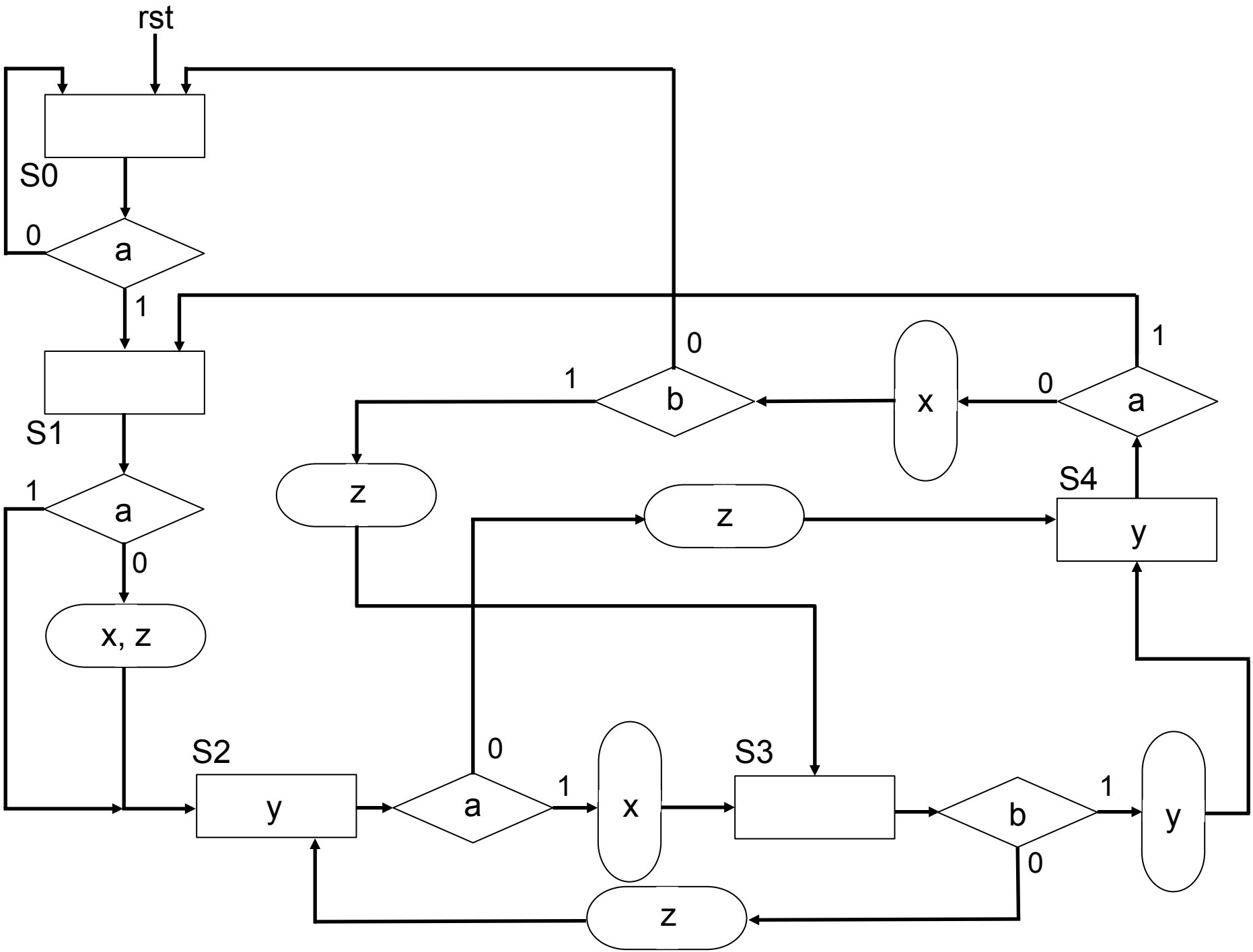
ECE 448
Midterm Exam
Solutions

Problem 1
FSM


rst
S0
0
a
1
S1
1
a
0
x, z
S2
y
0
a
1
x
z
z
0
b
1
x
S3
1
a
0
z
S4
y
1
b
0
y


rst
clk
a
b
S
x
y
z
0 ns
100 ns
200 ns
300 ns
400 ns
500 ns
600 ns


rst
clk
a
b
S
x
y
z
1
0
1
1
0
0
1
1
0
1
S0
S1
S2
S3
S2
S4
0 ns
100 ns
200 ns
300 ns
400 ns
500 ns
600 ns


rst
S0
0
a
1
S1
1
a
0
x, z
S2
y
0
a
1
x
z
z
0
b
1
x
S3
1
a
0
z
S4
y
1
b
0
y

Dataflow representation of the output function

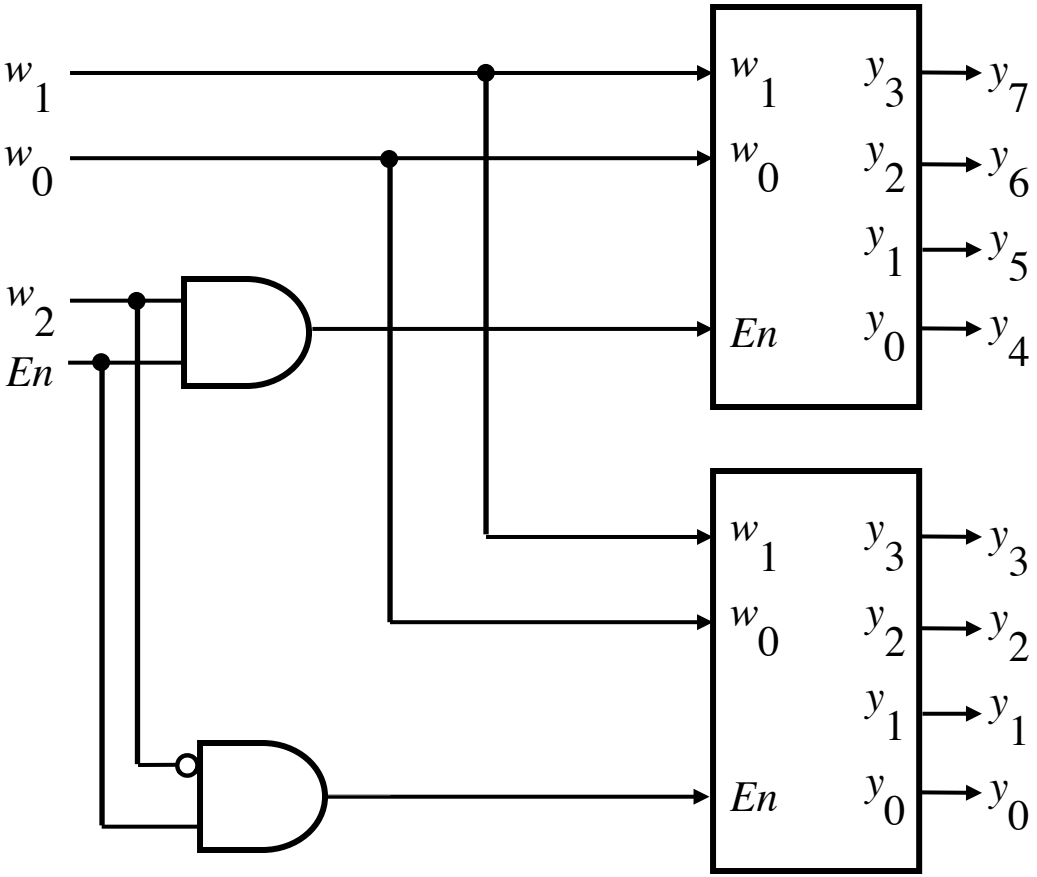
```
x <= '1' WHEN (State = S1 and a='0') OR (State = S2 and a='1')  OR
                (State = S4 and a='0') ELSE
     ’0';

y <= '1' WHEN (State = S2) OR (State = S3 and b='1') OR ( State = S4 )  ELSE
     '0';

z <= '1' WHEN (State = S1 and a='0') OR (State = S2 and a='0')  OR
                (State = S3 and b='0')  OR
                (State = S4 and a='0' and b='1') ELSE
     '0';
```

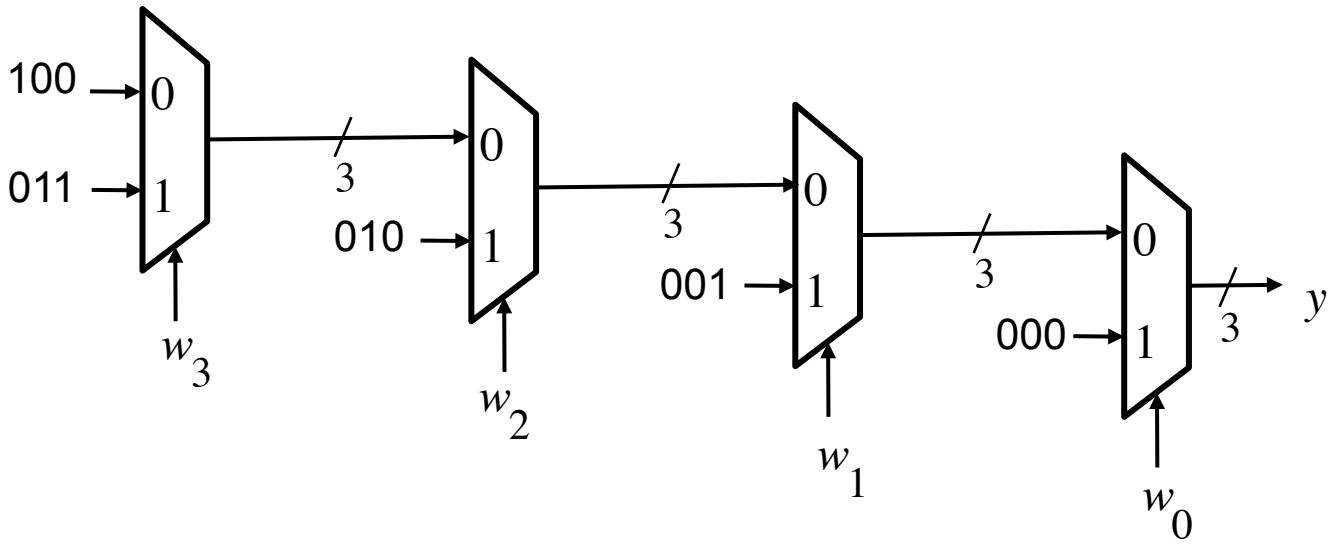
Problem 2
Block Diagrams
& Resource Utilization

A. 3-to-8 decoder (with enable) built of 2-to-4 decoders (with enable) and a minimum number of logic gates

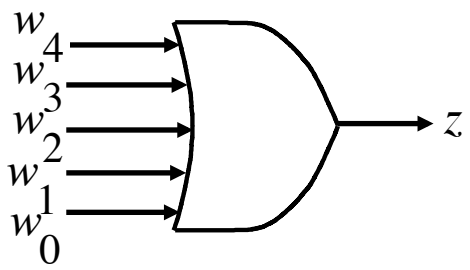


4 LUTs, ROM mode

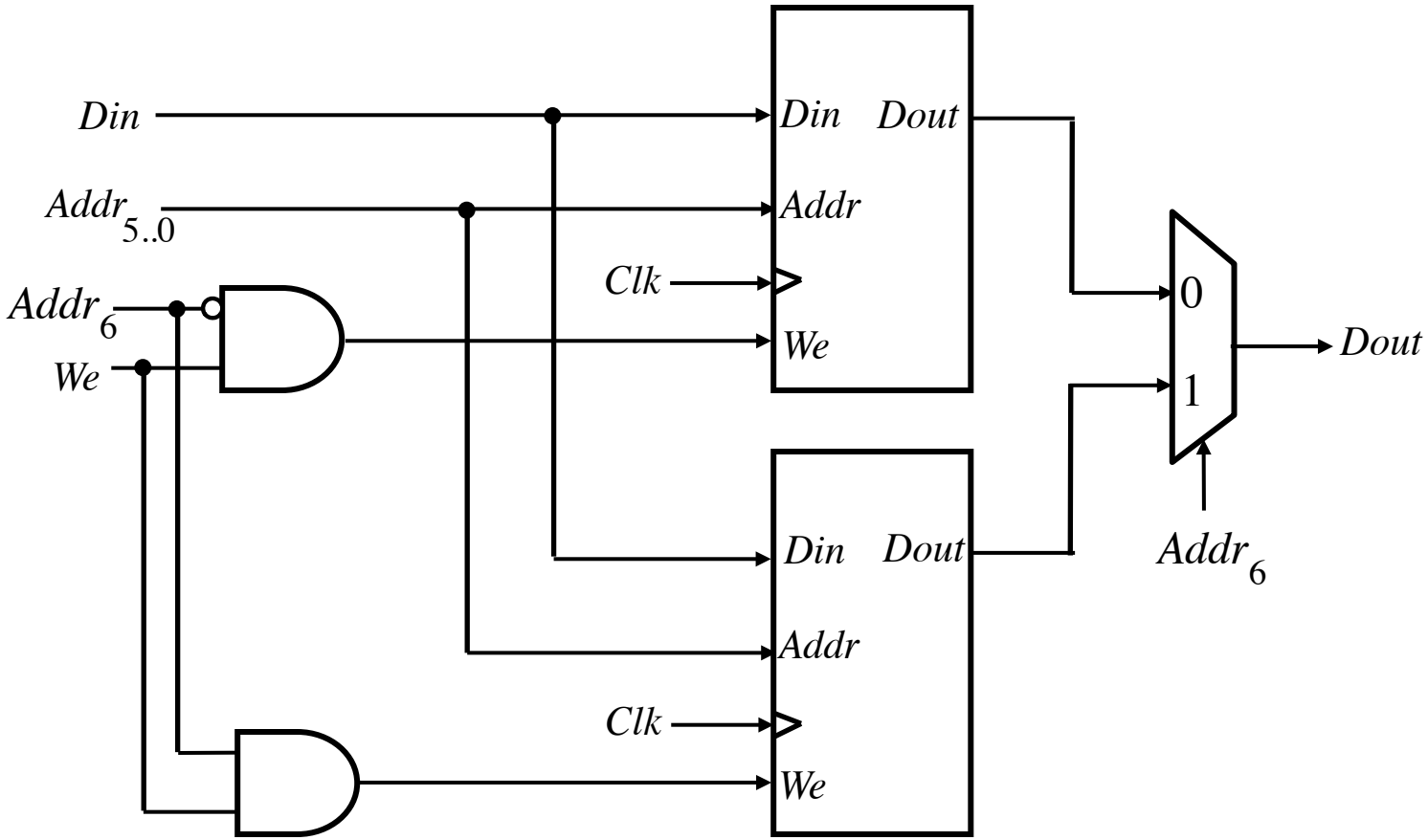
B. 5-bit priority encoder built of 2-to-1 multiplexers and a minimum number of logic gates (assume that the least significant bit of the input w has the highest priority, and the z output is equal to 1 when at least one input bit is equal to 1).



2 LUTs, ROM mode

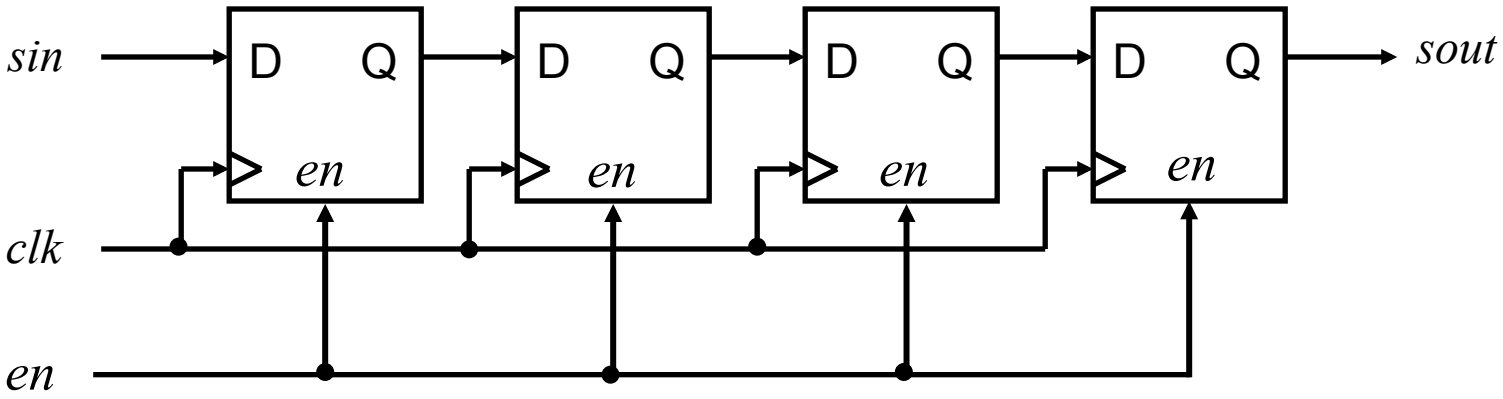


C. 128x1 RAM built of two 64x1 RAMs and a minimum number of logic gates



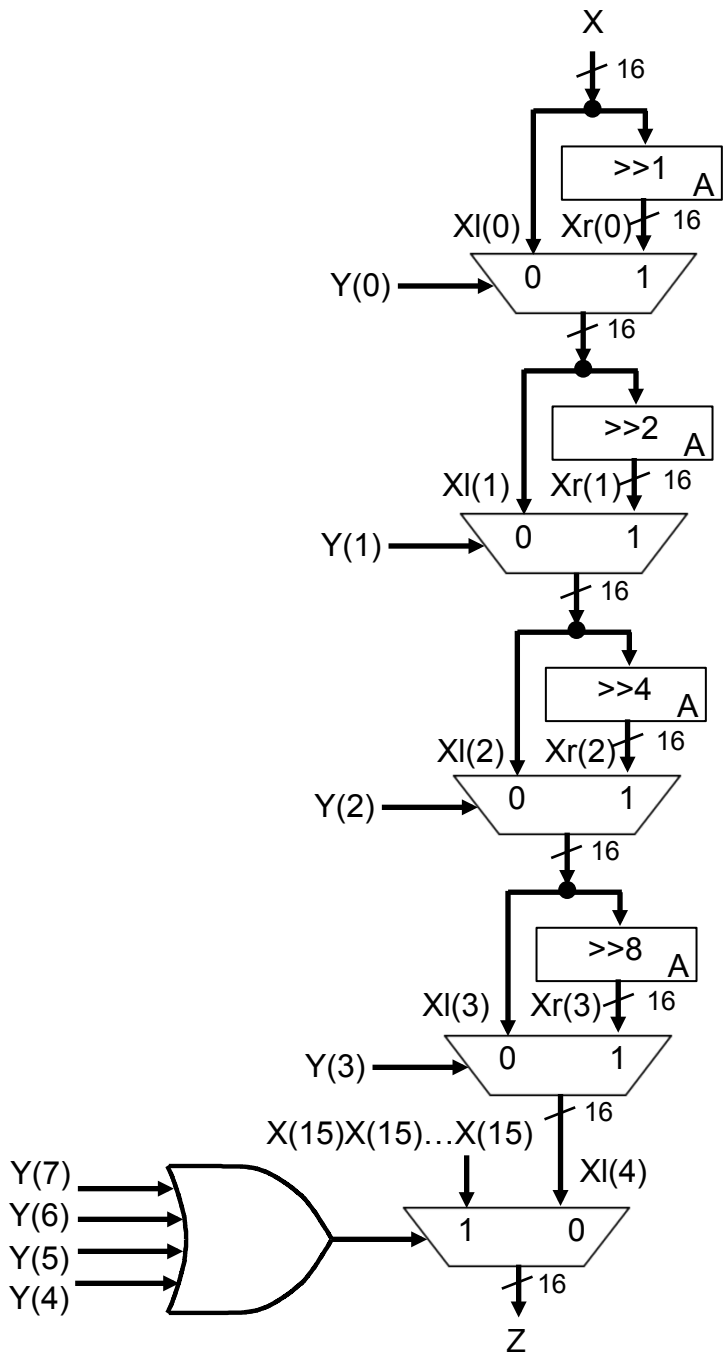
2 LUTs, RAM mode

D. 4-bit shift register with the serial input sin, enable en, clock clk, and the serial output sout.



1 LUT, SR mode

Problem 3
Variable Arithmetic Shifter Right


X
16
>>1
A
Xl(0)
Xr(0)
16
Y(0)
0
1
16
>>2
A
Xl(1)
Xr(1)
16
Y(1)
0
1
16
>>4
A
Xl(2)
Xr(2)
16
Y(2)
0
1
16
>>8
A
Xl(3)
Xr(3)
16
Y(3)
0
1
16
X(15)X(15)…X(15)
Xl(4)
Y(7)
Y(6)
Y(5)
Y(4)
1
0
16
Z

variable_shifter_right.vhd (1)

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;

ENTITY variable_shifter_right IS
        PORT(
                X : IN STD_LOGIC_VECTOR(15 downto 0);
                Y : IN STD_LOGIC_VECTOR(7 downto 0);
                Z : OUT STD_LOGIC_VECTOR(15 downto 0)
 );
END variable_shifter_right;

ARCHITECTURE mixed OF variable_shifter_right IS

TYPE array16 IS ARRAY (0 to 4) OF STD_LOGIC_VECTOR(15 downto 0);

SIGNAL Xl : array16;
SIGNAL Xr : array16;
```

variable_shifter_right.vhd (2)

```
BEGIN
        Xl(0) <= X;
        G: FOR i IN 0 TO 3 GENERATE

                SHIFT_I: ENTITY work.fixed_shifter_right(dataflow)
                        GENERIC MAP (L => 2** i)
                        PORT MAP ( a => Xl(i) ,
                                   y => Xr(i));

               Xl(i+1) <=  Xl(i) WHEN Y(i) = '0' ELSE Xr(i);

        END GENERATE;

        Z <= Xl(4) WHEN (Y(7) OR Y(6) OR Y(5) OR Y(4)) = '0' ELSE
             (OTHERS => X(15));

END mixed;
```

fixed_shifter_right.vhd

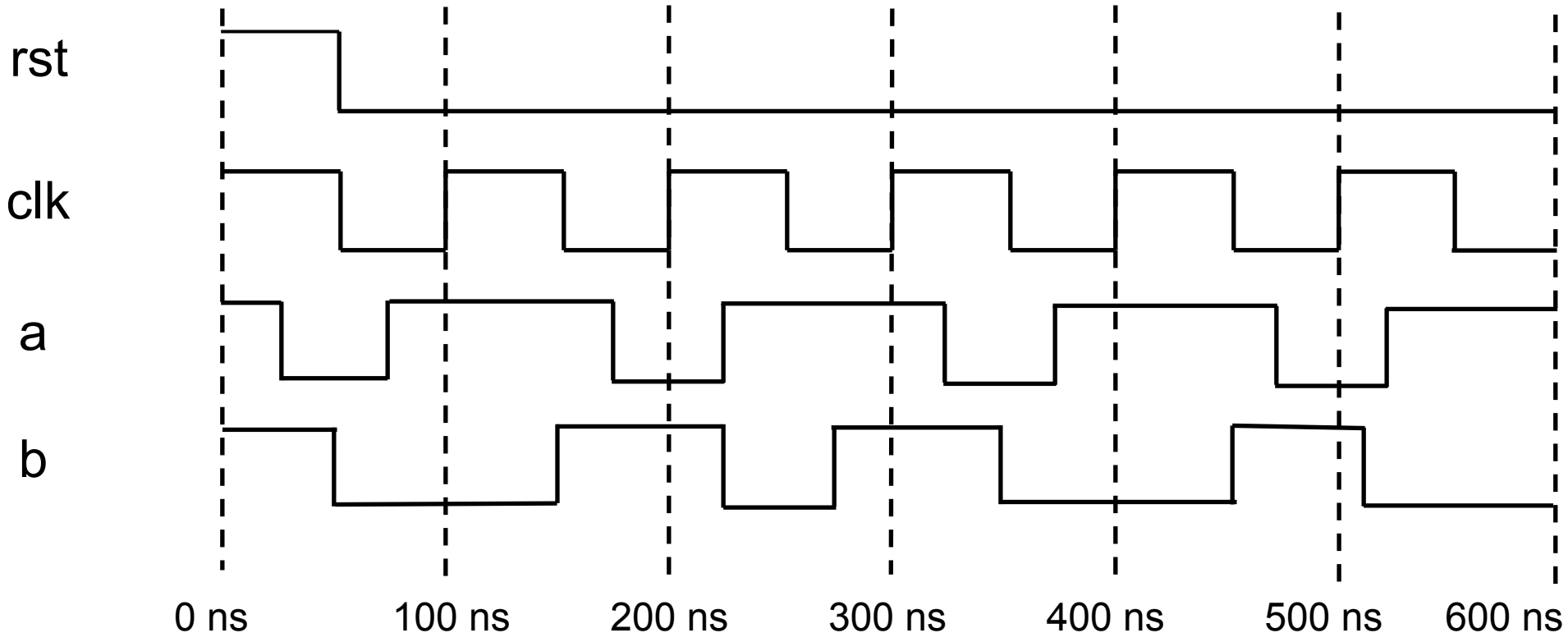
```
library IEEE;
use IEEE.STD_LOGIC_1164.all;

ENTITY fixed_shifter_right IS
        GENERIC (L : INTEGER := 1);
        PORT(
                a : IN STD_LOGIC_VECTOR(15 downto 0);
                y : OUT STD_LOGIC_VECTOR(15 downto 0)
          );

END fixed_shifter_right;

ARCHITECTURE dataflow OF fixed_shifter_right IS
BEGIN
        y(15-L downto 0) <= a(15 downto L);
        y(15 downto 15-L+1) <= (OTHERS => a(15));
END dataflow;
```

Problem 4
Testbench


rst
clk
a
b
0 ns
100 ns
200 ns
300 ns
400 ns
500 ns
600 ns

controller_tb.vhd (1)

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;

ENTITY controller_tb IS
END controller_tb;

ARCHITECTURE behavioral OF controller_tb IS

SIGNAL clk: STD_LOGIC := '1’;
SIGNAL rst: STD_LOGIC := '1';
SIGNAL a : STD_LOGIC := '1';
SIGNAL b : STD_LOGIC := '1';
SIGNAL x, y, z: STD_LOGIC;

CONSTANT clk_period : TIME := 100 ns;
CONSTANT reset_length : TIME := clk_period/2;
```

controller_tb.vhd (2)

```
BEGIN
        UUT: ENTITY work.Controller(mixed)
                PORT MAP (clk => clk,
                rst => rst,
                a => a,
                b => b,
                x => x,
                y => y,
                z => z);

        clk <= NOT clk AFTER clk_period/2;

        PROCESS
        BEGIN
                WAIT FOR reset_length;
                rst <= '0';
                WAIT;
        END PROCESS;
```

controller_tb.vhd (3)

```
PROCESS
BEGIN
        WAIT FOR clk_period/4;
        a <= '0';
        WAIT FOR clk_period/2;
        a <= '1';
        WAIT FOR 3*clk_period/4;
END PROCESS;
```

controller_tb.vhd (3)

```
        PROCESS
        BEGIN
                WAIT FOR clk_period/2;
                b <= '0';
                WAIT FOR clk_period;
                b <= '1';
                WAIT FOR 3*clk_period/4;
                b <= '0';
                WAIT FOR clk_period/2;
                b <= '1';
                WAIT FOR 3*clk_period/4;
                b <= '0';
                WAIT FOR clk_period;
                b <= '1';
                WAIT FOR 5*clk_period/8;
                b <= '0';
                WAIT;
        END PROCESS;
END behavioral;
```