

# LAB4 GRP7 SESS202 REPORT

## VHDL SOURCE CODE 1: TOP LEVEL STRUCTURAL

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

USE ieee.numeric\_std.ALL;

ENTITY LogicalStep\_Lab4\_top IS

PORT

(

clkin\_50 : in std\_logic;

rst\_n : in std\_logic;

pb : in std\_logic\_vector(3 downto 0);

sw : in std\_logic\_vector(7 downto 0); -- The switch inputs

leds : out std\_logic\_vector(7 downto 0); -- for displaying the switch content

seg7\_data : out std\_logic\_vector(6 downto 0); -- 7-bit outputs to a 7-segment

seg7\_char1 : out std\_logic; -- seg7 digi selectors

seg7\_char2 : out std\_logic -- seg7 digi selectors

);

END LogicalStep\_Lab4\_top;

ARCHITECTURE SimpleCircuit OF LogicalStep\_Lab4\_top IS

COMPONENT SevenSegment is port (

hex : in std\_logic\_vector(3 downto 0); -- The 4 bit data to be displayed

sevenseg : out std\_logic\_vector(6 downto 0) -- 7-bit outputs to a 7-segment

);

end COMPONENT;

COMPONENT segment7\_mux is

port (

clk : in std\_logic := '0';

DIN2 : in std\_logic\_vector(6 downto 0);

DIN1 : in std\_logic\_vector(6 downto 0);

DOUT : out std\_logic\_vector(6 downto 0);

DIG2 : out std\_logic;

DIG1 : out std\_logic

);

end COMPONENT;

COMPONENT MealyStatemachine IS

Port(

clk\_input, resetButton : IN std\_logic;

X\_EQ, X\_GT, X\_LT : IN std\_logic; --comparing DESIRED TO ACTUAL

Y\_EQ, Y\_GT, Y\_LT : IN std\_logic; --comparing DESIRED TO ACTUAL

extenderOut : IN std\_logic;

X\_MOTION, Y\_MOTION : IN std\_logic;

X\_Clk\_en, X\_UPorDOWN: OUT std\_logic;

Y\_Clk\_en, Y\_UPorDOWN: OUT std\_logic;

ExtenderEnable: OUT std\_logic;

isError : OUT std\_logic

);

END COMPONENT;

COMPONENT Bin\_Counter4bit is port

(

Main\_clk : in std\_logic;

rst\_n : in std\_logic := '0';

clk\_en : in std\_logic := '0';

up1\_down0 : in std\_logic := '0';

counter\_bits : out std\_logic\_vector(3 downto 0)

);

end COMPONENT;

COMPONENT FourBitComparator is port (

bitA0, bitA1, bitA2, bitA3, bitB0, bitB1, bitB2, bitB3 : in std\_logic;

AGTB : out std\_logic;

AEQB : out std\_logic;

ALTB : out std\_logic

);

end COMPONENT;

COMPONENT Grappler IS Port (

clk\_input, rst\_n, controlButton, enable : IN std\_logic;

grappleControl : OUT std\_logic

);

end COMPONENT;

COMPONENT Bidir\_shift\_reg is port

(

CLK : in std\_logic :='0';

RESET\_n : in std\_logic :='0';

CLK\_EN : in std\_logic :='0';

LEFT0\_RIGHT1 : in std\_logic :='0';

REG\_BITS : OUT std\_logic\_vector(3 downto 0)

);

end COMPONENT;

COMPONENT Extender IS Port

(

clk\_input, rst\_n, controlButton, enable : IN std\_logic;

currentShiftValue : IN std\_logic\_vector(3 downto 0);

bitShifting, extenderOut, bitShiftDirection, grapplerEnable : OUT std\_logic

);

END COMPONENT;

COMPONENT mux is port (

hex\_in1, hex\_in2 : in std\_logic\_vector(6 downto 0);

mux\_select :in std\_logic;

hex\_out : out std\_logic\_vector(6 downto 0)

);

end COMPONENT;

COMPONENT muxSingle is port (

hex\_in1, hex\_in2 : in std\_logic;

mux\_select :in std\_logic;

hex\_out : out std\_logic

);

end COMPONENT;

COMPONENT FlashCounter is port

(

Main\_clk : in std\_logic;

rst\_n : in std\_logic := '0';

clk\_en : in std\_logic := '0';

up1\_down0 : in std\_logic := '0';

counter\_bits : out std\_logic

);

end COMPONENT;

----------------------------------------------------------------------------------------------------

CONSTANT SIM : boolean := FALSE; -- set to TRUE for simulation runs otherwise keep at 0.

CONSTANT CLK\_DIV\_SIZE : INTEGER := 26; -- size of vectors for the counters

SIGNAL Main\_Clk : STD\_LOGIC; -- main clock to drive sequencing of State Machine

SIGNAL bin\_counter : UNSIGNED(CLK\_DIV\_SIZE-1 downto 0); -- := to\_unsigned(0,CLK\_DIV\_SIZE); -- reset binary counter to zero

----------------------------------------------------------------------------------------------------

SIGNAL XTARGET7seg, YTARGET7seg : std\_logic\_vector(6 downto 0);

SIGNAL XCURRENT, YCURRENT : std\_logic\_vector(3 downto 0);

SIGNAL XCURRENT7seg, YCURRENT7seg : std\_logic\_vector(6 downto 0);

SIGNAL XMUX7seg, YMUX7seg: std\_logic\_vector(6 downto 0);

SIGNAL XTargLTCurr, XTargEQCurr, XTargGTCurr : std\_logic;

SIGNAL YTargLTCurr, YTargEQCurr, YTargGTCurr : std\_logic;

SIGNAL bitShiftDirControl, bitShiftEnable : std\_logic;

SIGNAL currentShiftValue : std\_logic\_vector(3 downto 0);

SIGNAL extenderOutSignal : std\_logic;

SIGNAL X\_ClockEnable, Y\_ClockEnable : std\_logic;

SIGNAL X\_Direction, Y\_Direction : std\_logic;

SIGNAL GrappleEnableSignal : std\_logic;

SIGNAL extenderEnableSignal : std\_logic;

SIGNAL ERROR7seg, NOTHING7seg, ERROR7segOutput : std\_logic\_vector(6 downto 0);

SIGNAL X\_ERROR\_AND\_VALUE7seg, Y\_ERROR\_AND\_VALUE7seg:std\_logic\_vector(6 downto 0);

SIGNAL ERROR : std\_logic;

SIGNAL MUX\_CLOCK : std\_logic;

SIGNAL FLASH: std\_logic;

----------------------------------------------------------------------------------------------------

BEGIN

-- CLOCKING GENERATOR WHICH DIVIDES THE INPUT CLOCK DOWN TO A LOWER FREQUENCY

BinCLK: PROCESS(clkin\_50, rst\_n) is

BEGIN

IF (rising\_edge(clkin\_50)) THEN -- binary counter increments on rising clock edge

bin\_counter <= bin\_counter + 1;

END IF;

END PROCESS;

Clock\_Source:

Main\_Clk <=

clkin\_50 when sim = TRUE else -- for simulations only

std\_logic(bin\_counter(23)); -- for real FPGA operation

---------------------------------------------------------------------------------------------------

leds (7 downto 4) <= currentShiftValue (3 downto 0);

leds (0) <= ERROR;

ERROR7seg <= "1111001";

NOTHING7seg <= "0000000";

---------------------------------------------------------------------------------------------------

INST1XTARGET: SevenSegment PORT MAP (sw(7 downto 4),XTARGET7Seg);

INST2YTARGET: SevenSegment PORT MAP (sw(3 downto 0),YTARGET7Seg);

INST11XCURRENT: SevenSegment PORT MAP (XCURRENT,XCURRENT7seg);

INST12YCURRENT: SevenSegment PORT MAP (YCURRENT,YCURRENT7seg);

INST13MUX\_X:mux PORT MAP (

XCURRENT7seg, XTARGET7Seg,

pb(3),

XMUX7seg);

INST14MUX\_Y:mux PORT MAP (

YCURRENT7seg, YTARGET7Seg,

pb(2),

YMUX7seg);

INST15X\_MUX\_ERROR :mux PORT MAP (

XMUX7seg, ERROR7segOutput,

ERROR,

X\_ERROR\_AND\_VALUE7seg);

INST16Y\_MUX\_ERROR :mux PORT MAP (

YMUX7seg,ERROR7segOutput,

ERROR,

Y\_ERROR\_AND\_VALUE7seg);

INST18ERROR\_AND\_NOTHING: mux PORT MAP(

ERROR7seg, NOTHING7seg,

FLASH,

ERROR7segOutput);

INST19FLASH: FlashCounter PORT MAP

(

std\_logic(bin\_counter(23)),

rst\_n,

'1',

'1',

FLASH

);

INST3: segment7\_mux PORT MAP (clkin\_50, X\_ERROR\_AND\_VALUE7seg(6 downto 0), Y\_ERROR\_AND\_VALUE7seg(6 downto 0), seg7\_data(6 downto 0), seg7\_char1, seg7\_char2);

INST4X: Bin\_Counter4bit PORT MAP (Main\_Clk,rst\_n,X\_ClockEnable,X\_Direction,XCURRENT(3 downto 0) );

INST5Y: Bin\_Counter4bit PORT MAP (Main\_Clk,rst\_n, Y\_ClockEnable,Y\_Direction,YCURRENT(3 downto 0) );

INST6X: FourBitComparator PORT MAP (

sw(4),sw(5), sw(6), sw(7),

XCURRENT(0), XCURRENT(1), XCURRENT(2), XCURRENT(3),

XTargGTCurr, XTargEQCurr, XTargLTCurr);

INST7Y: FourBitComparator PORT MAP (

sw(0),sw(1), sw(2), sw(3),

YCURRENT(0), YCURRENT(1), YCURRENT(2), YCURRENT(3),

YTargGTCurr, YTargEQCurr, YTargLTCurr);

INST8GrapplerSM: Grappler PORT MAP (Main\_Clk, rst\_n, pb(0) , GrappleEnableSignal, leds(3));

INST9: Bidir\_shift\_reg PORT MAP(

Main\_Clk, rst\_n,

bitShiftEnable, bitShiftDirControl,

currentShiftValue (3 downto 0) );

--

INST10: Extender PORT MAP (

Main\_Clk, rst\_n, pb(1), extenderEnableSignal,

currentShiftValue(3 downto 0),

bitShiftEnable, extenderOutSignal, bitShiftDirControl,GrappleEnableSignal

);

INSTMEALY: MealyStatemachine PORT MAP (

Main\_Clk, rst\_n,

XTargEQCurr, XTargGTCurr, XTargLTCurr,

YTargEQCurr, YTargGTCurr, YTargLTCurr,

extenderOutSignal,

pb(3), pb(2),

X\_ClockEnable, X\_Direction,

Y\_ClockEnable, Y\_Direction,

extenderEnableSignal,

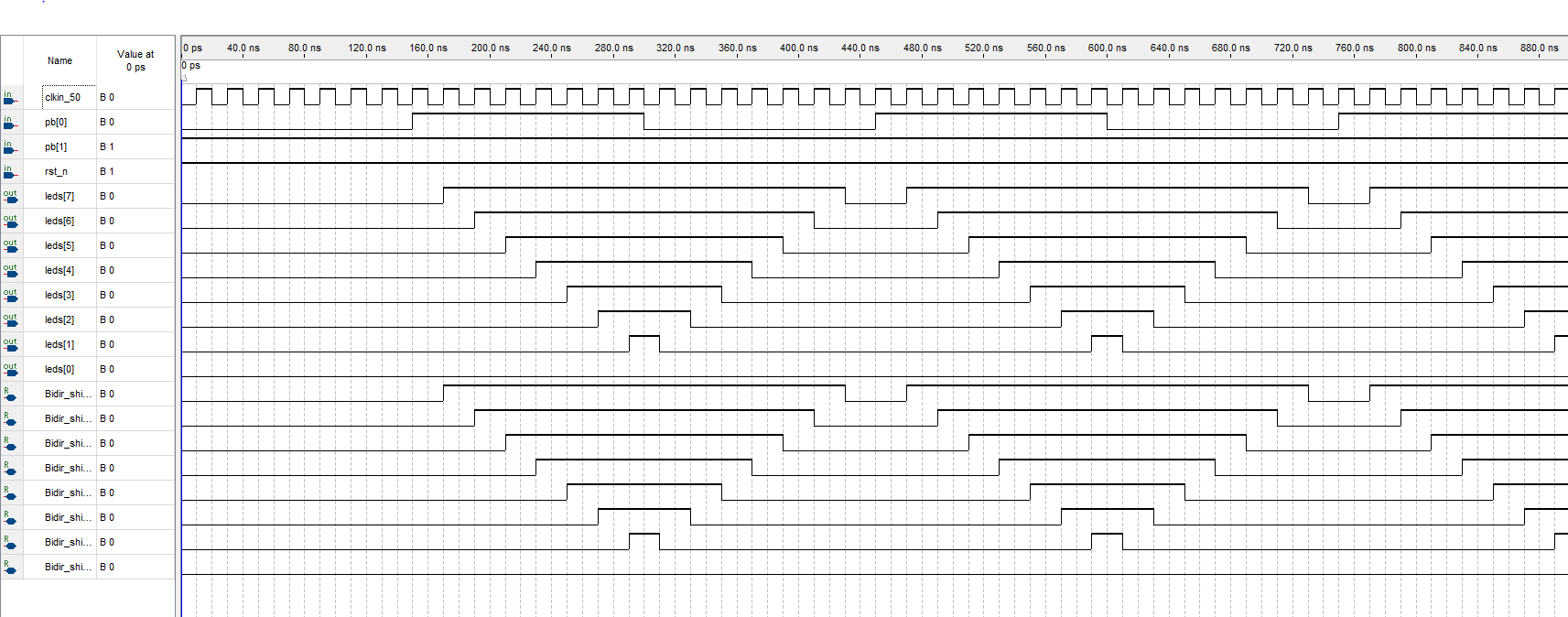
ERROR

);

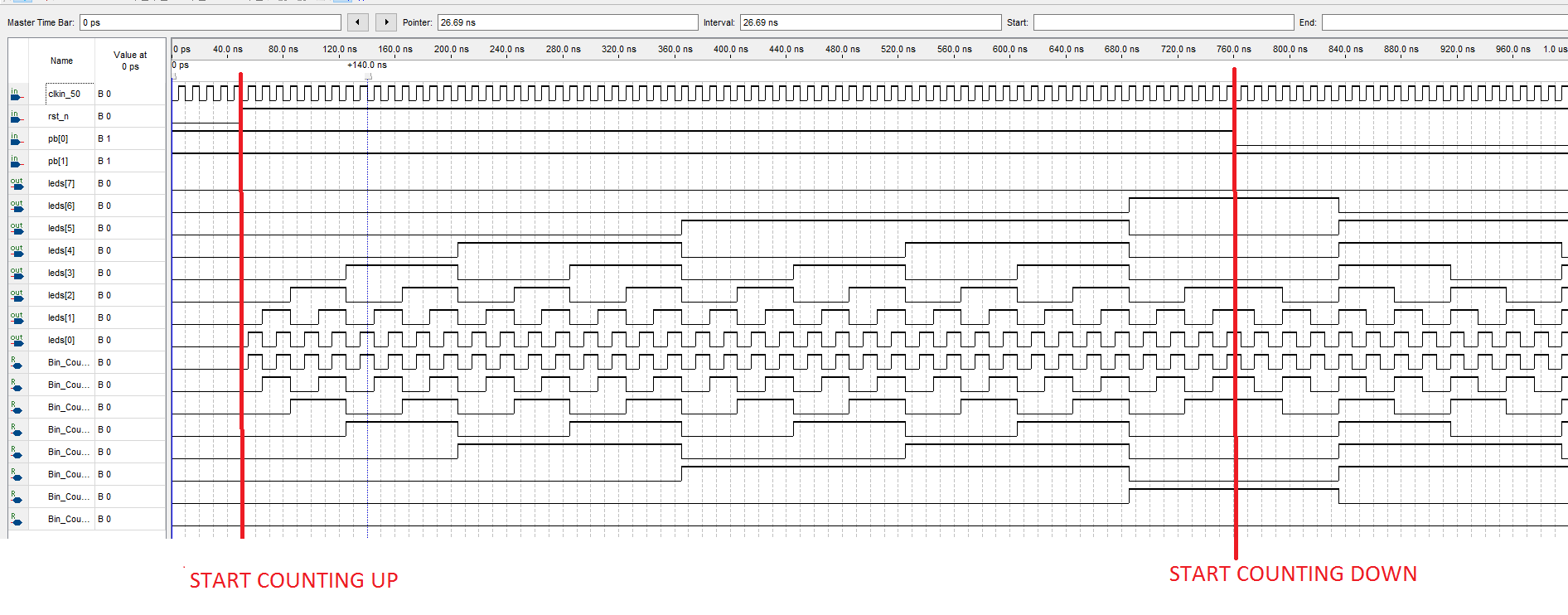
--INST13ERROR\_MUX: mux PORT MAP (ERROR7seg,

END SimpleCircuit;

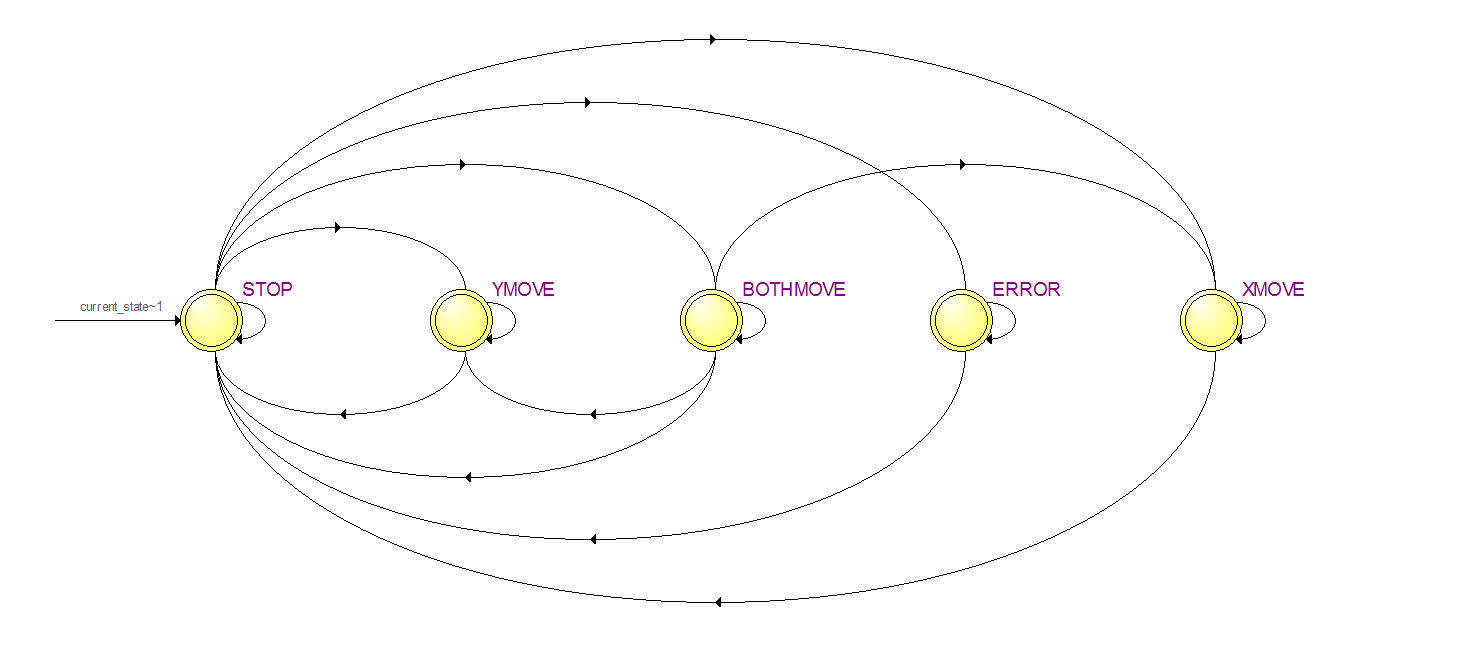
## Simulation 1: 8-Bits Shift Register



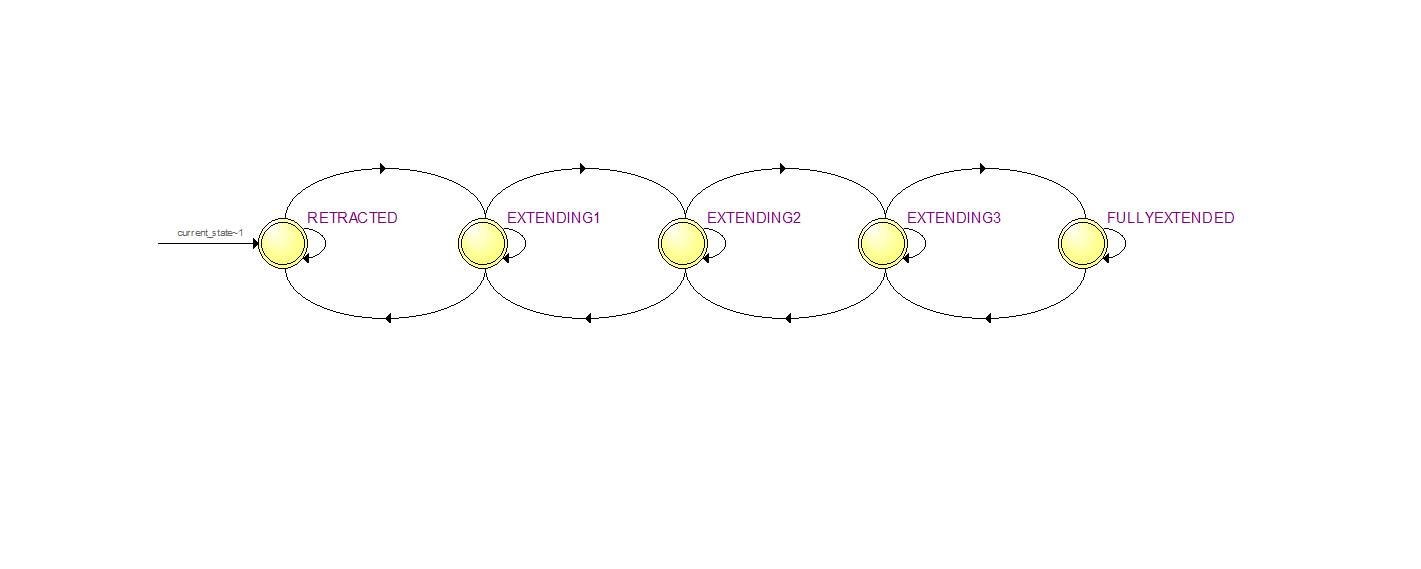
## Simulation 2: 8-Bits Counter



## State Diagram 1 Mealy State Machine



## State Diagram 2: Extender Moore State Machine



## State Diagram 3: Grappler Moore State Machine

## Fitter Report on Resources Utilization by Entity

