Senior Project Design Review:

Internal Hardware Design of a Microcontroller in VLSI

Designers:

Shreya Prasad & Heather Smith

Advisor:

Dr. Vinod Prasad

March 11th, 2003

Presentation Outline

- Project summary
- Review of preliminary work
- Project description (by subsystems)
  - Functional description
  - Block diagram
- Schedule of tasks

Project Summary

•To design some of the internal components of a microcontroller using L-EDIT.

•To create several delays by designing the internal 16-bit timer circuitry and 4 registers.

• These delays can run real time systems and interrupts.

•The design will also be done in VHDL code and implemented into FPGA.

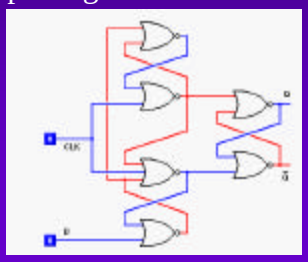
Preliminary Work

•The microcontroller to be designed is similar to the Motorola 68HC11.

•The designers used Motorola's microcontroller as a guide; however, their logic design was independent.

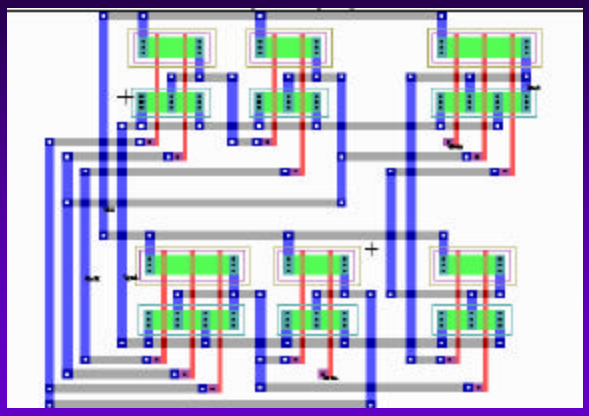
•*http://www.play-hookey.com* helped the designers with the gate level design for the D flip-flop and the XOR gate.

•The D flip-flop designed used the least number of gates.



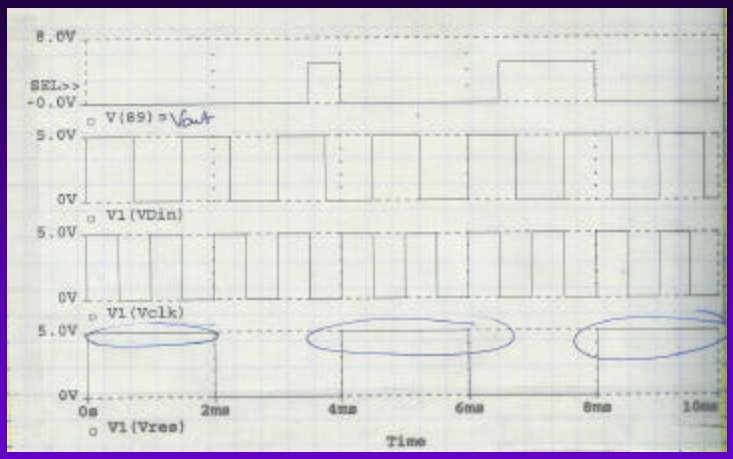
Preliminary Work

D flip-flop in LEDIT



Project Description

D flip-flop plot in PSPICE



Project Description

Functional Description

User inputs:

- 8-bit accumulator
- 2-bit register controller input and 1-bit enable
- 2-bit clock controller input and 1-bit enable
- timer reset
- clock

User outputs:

- 1 bit from each of the four comparators
- 1 bit overflow from timer

Project Description

Functional Description

Modes of operation: The user inputs 8 bits into the accumulator and 2 bits to the register controller and clock controller. The register controller specifies which register to store the 8 bits into. The clock controller specifies the period of the input clock pulse.

The 16-bit timer continuously increments with the clock. The lower 8 bits input into each of the 4 comparators. They are then compared to the 8 bits stored in each register. The comparator output is logic ‘1’ when the two values are equal.

Project Description

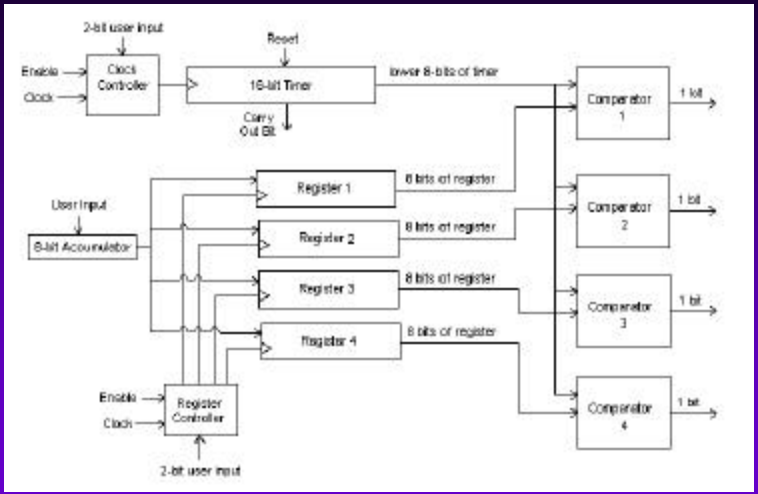
Block Diagram

Subsystems:

- Clock Controller
- 16-bit Timer
- Register Controller
- Register
- Comparator

Project Description

System Block Diagram



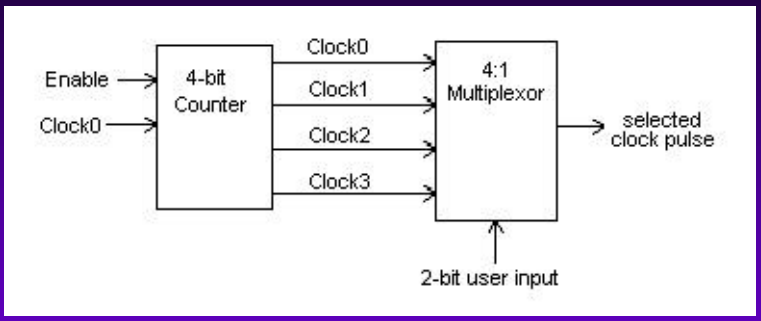
Project Description

Clock Controller Subsystem

- Inputs:
  - Clock (user input)
  - Enable (user input)
  - 2-bit Control Value (user input)
- Outputs:
  - Clock (to timer and register controller subsystems)

Project Description

Clock Controller Subsystem Block Diagram



Project Description

Description of Clock Controller Subsystem

•This controls the clock pulse entering the timer subsystem.

•The user inputs the fastest clock pulse desired into the clock input.

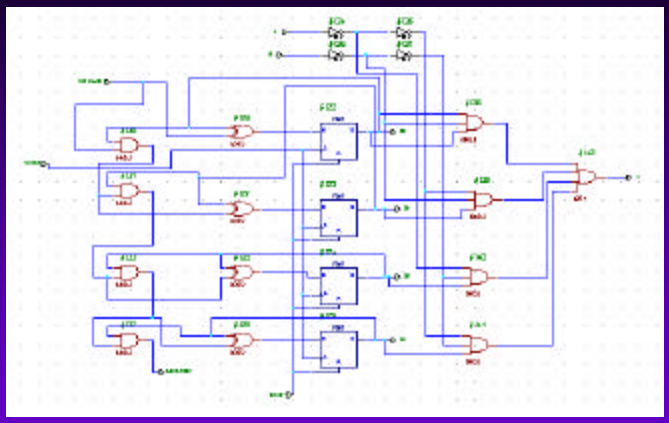
•The subsystem can then output that pulse, or a pulse two times, four times, or eight times the period of the original pulse, into the timer subsystem.

•The 2-bit controller value selects which pulse to output using a 4 to 1 multiplexer.

•The pulse periods are decreased using a 4-bit counter.

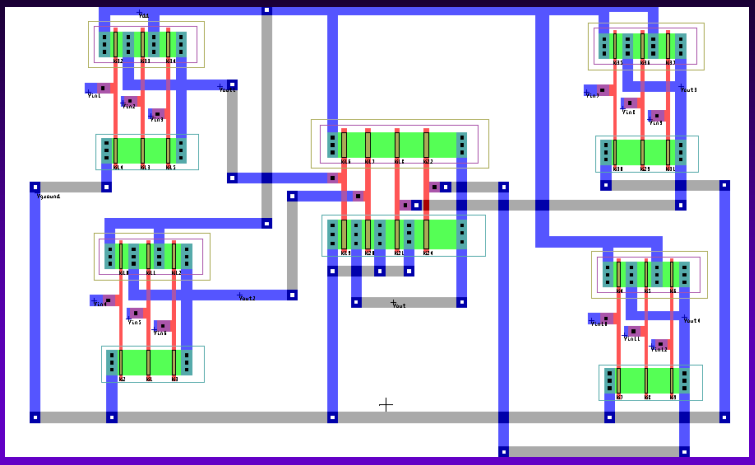
Project Description

Xilinx Design of Clock Controller Subsystem



Project Description

LEDIT Design of Clock Controller Subsystem



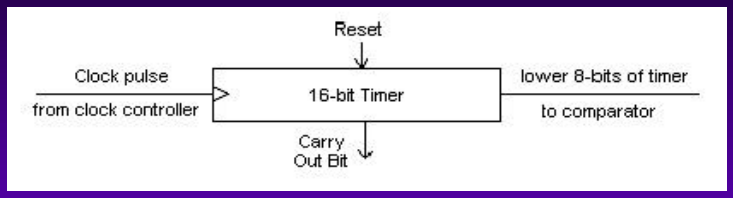
Project Description

16-bit Timer Subsystem

- Inputs:
  - Clock (from clock controller subsystem)
  - Enable (user input)
  - Reset (user input)
- Outputs:
  - Lower 8 bits of timer value (to comparator subsystem)
  - Overflow bit (external output)

Project Description

16-bit Timer Subsystem Block Diagram



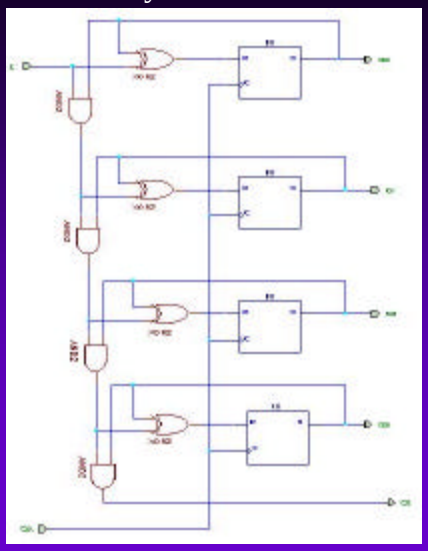
Project Description

Description of Timer Subsystem

- This is the main subsystem in the project.
- The enable input starts and stops the counter.
- It operates as a 16-bit counter. After reaching FFFFh, it sets the overflow bit high for one clock pulse and will restart at 0000h.
- Although only the lower 8 bits are used in the timer, having 16 bits will allow for longer delays.

Project Description

Xilinx of Timer Subsystem



Project Description

LEDIT of Timer Subsystem

Project Description

VHDL Design of Timer Subsystem

```
library ieee;
use ieee.std_logic_1164.all;
entity timer16 is
        port(enable,clk,rst: in std_logic;
        co: out std_logic;
        qout: out integer range 0 to 15);
end timer16;
architecture rtl of timer16 is
signal count: integer range 0 to 15;
begin
        ctr: process (enable,clk,rst)
        begin
                if (rst = '1') then
                co <= '0';
                count <= 0;
                elsif (clk'event and clk = '1' and enable = '1') then
                        if (count >= 15) then
                        co <= '0';
                        count <= 0;
                        else
                        co <= '0';
                        count <= count+1;
                                if count = 15 then
                                co <= '1';
                                end if;
                        end if;
                end if;
        end process;
        qout <= count;
end rtl;
```

Project Description

Register Controller Subsystem

Inputs:

Enable (user input)

Clock (from clock controller subsystem)

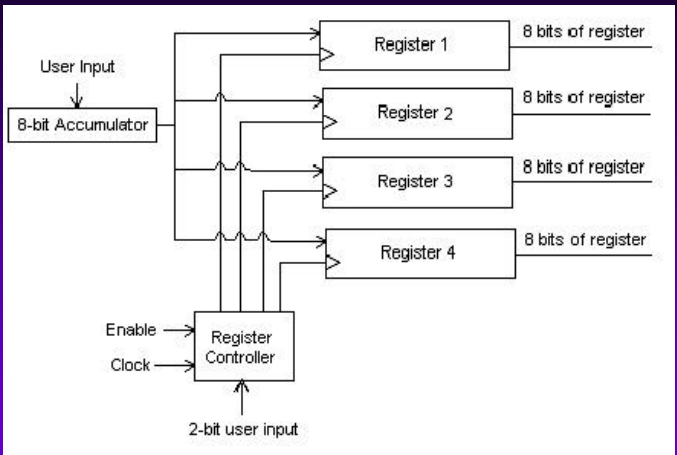
2-bit Control Value (user input)

Outputs:

Clock (to register subsystem)

Project Description

Register Controller Subsystem Block Diagram



Project Description

Description of Register Controller Subsystem.

•The 2-bit controller value specifies which of the 4 registers to point the 8-bit accumulator value to.

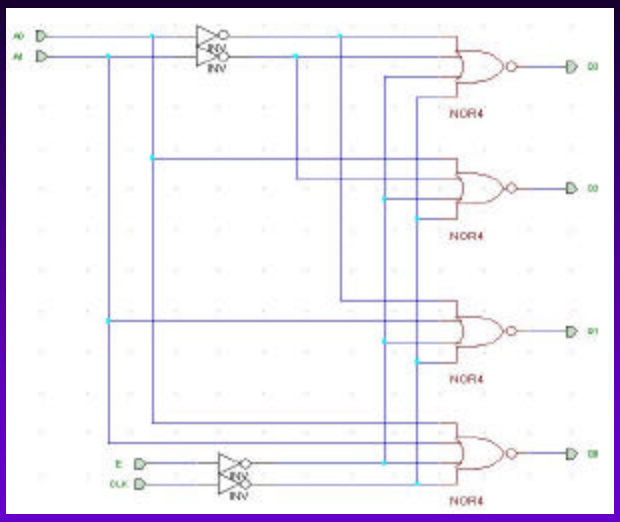
•A register only receives the accumulator value when the controller turns its clock on.

•Only register clock can be on at a time.

•A 1 to 4 decoder is used to select which register will receive the input clock pulse.

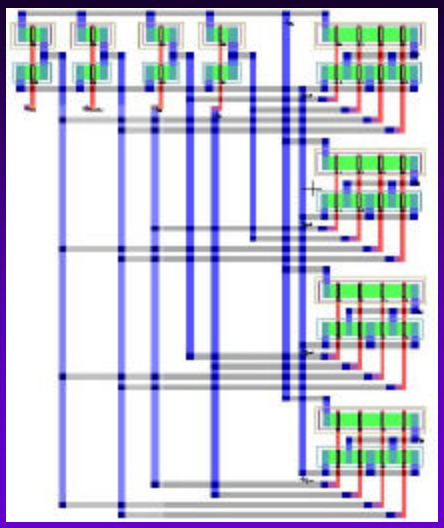
Project Description

Xilinx of Register Controller Subsystem



Project Description

LEDIT of Register Controller Subsystem



Project Description

VHDL Design of Register Controller Subsystem

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

entity regcon is
        port(a: in std_logic_vector (1 downto 0);
        enable,clk: in std_logic;
        q: out std_logic_vector (3 downto 0));
end regcon;

architecture rtl of regcon is
component not1
        port (x: in std_logic;
        z: out std_logic);
end component;

component nor4
        port (a,b,c,d: in std_logic;
        output: out std_logic);
end component;

signal nclk,nenable: std_logic;
signal na: std_logic_vector (1 downto 0);

begin
        g1: not1 port map (a(0),na(0));
        g2: not1 port map (a(1),na(1));
        g3: not1 port map (clk,nclk);
        g4: not1 port map (enable,nenable);
        g5: nor4 port map (na(0),na(1),nclk,nenable,q(0));
        g6: nor4 port map (a(0),na(1),nclk,nenable,q(2));
        g7: nor4 port map (na(0),a(1),nclk,nenable,q(1));
        g8: nor4 port map (a(0),a(1),nclk,nenable,q(3));
end rtl;
```

Project Description

Register Subsystem

Inputs:

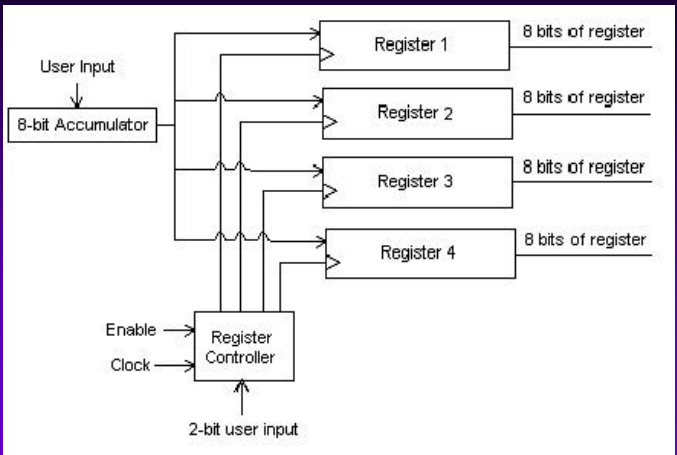
- 8-bit Accumulator Value (from accumulator)
- Clock (from register controller)

Outputs:

- 8-bit Register Value (to comparator subsystem)

Project Description

Register Subsystem Block Diagram



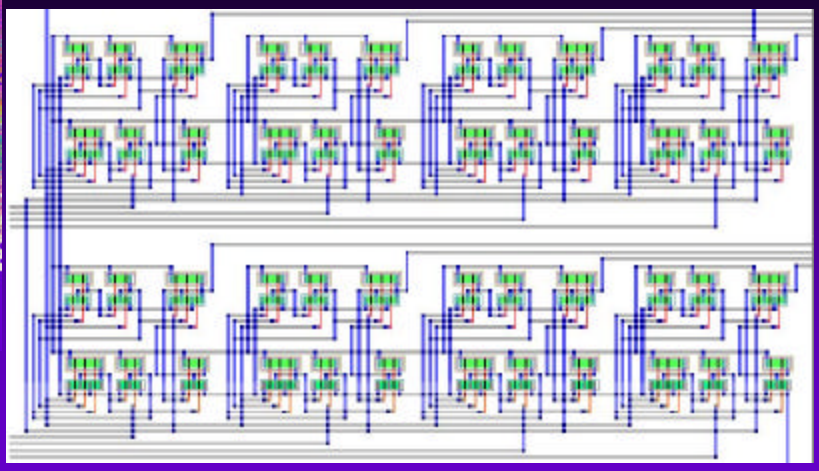
Project Description

Description of Register Subsystem

- Each register will consist of 8 D flip-flops, which will store the register value.
- A register stores the accumulator value only when the controller turns on its clock.
- Otherwise, the register will hold its previous value until this value is written over with a new value from the accumulator.
- The inputs and outputs previously mentioned are for each of the 4 registers.

Project Description

LEDIT of Register Subsystem



Project Description

VHDL Code for Register Subsystem

- A D flip-flop component was written in a program.
- A program for the register was written. It includes port maps for each of the 8-bits.
- Four 8-bit registers were needed, so another program was written using port maps, as shown here:

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

entity reg is
port(D1,D2,D3,D4:in std_logic_vector(7 downto 0);
        clk,rst: in std_logic;
        Q1,Q2,Q3,Q4:out std_logic_vector(7 downto 0));
end reg;

architecture arch of reg is
component reg8
        port(D:in  std_logic_vector(7  downto  0);
        clk,rst: in std_logic;
        Q: out std_logic_vector(7 downto 0));
end component;

begin
        g9:reg8 port map(D=>D1,clk=>clk,rst=>rst,Q=Q1);
        g10:reg8 port map(D=>D2,clk=>clk,rst=>rst,Q=Q2);
        g11:reg8 port map(D=>D3,clk=>clk,rst=>rst,Q=Q3);
        g12:  reg8 port map(D=>D4,clk=>clk,rst=>rst,Q=Q4);
end arch;
```

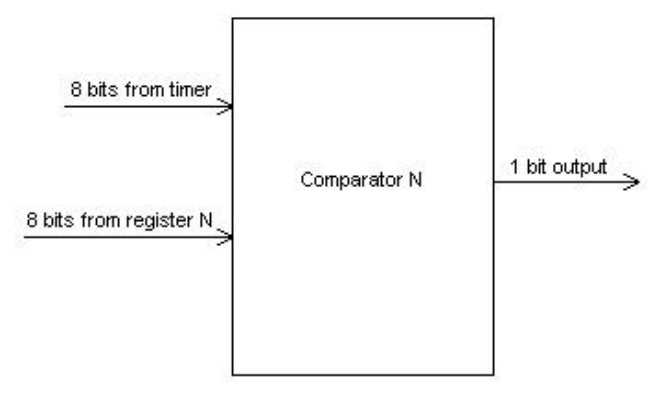
Project Description

Comparator Subsystem

- Inputs:
  - 8-bit Register Value (from register subsystem)
  - Lower 8-bit Timer Value (from timer subsystem)
- Outputs:
  - 1-bit Compared Value (external output)

Project Description

Comparator Subsystem Block Diagram



Project Description

Description of Comparator Subsystem

•There are 4 comparators, 1 for each register.

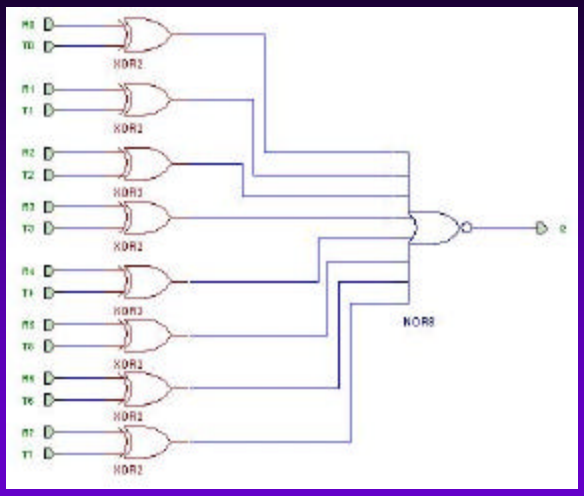
•The 8-bit value from each register is compared to the lower 8-bits of the timer.

•If the two inputs are the same, the output is logic ‘1’.

•Otherwise, the output is logic ‘0’.

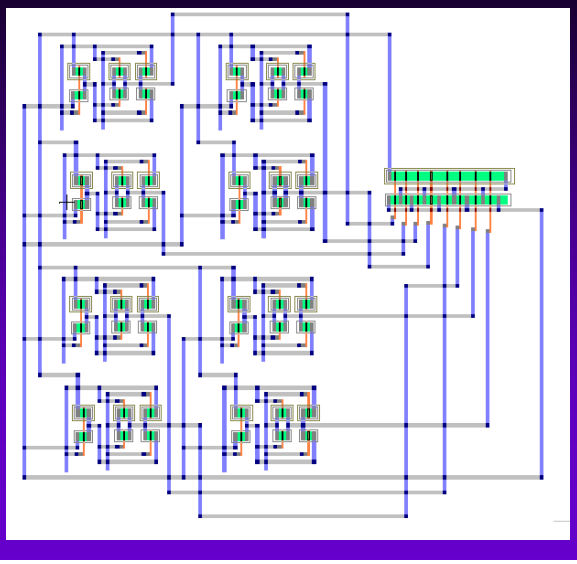
Project Description

Xilinx of Comparator Subsystem



Project Description

LEDIT of Comparator Subsystem



Project Description

VHDL Code for Comparator Subsystem

- An XOR gate was written in a program.
- A NOR-8 gate was written in a program.
- The comparator program declared xor_gate and nor8_gate as components. Port maps were used as shown below:

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

entity comp is
port(r,t: in std_logic_vector(7 downto 0);
        z:out std_logic);
end comp;

architecture sgp of comp is
component xor_gate is
port(a,b: in std_logic;
z:out std_logic);
end component;

component nor8_gate
port(input: in std_logic_vector(7 downto 0);
        output:out std_logic);
end component;

signal s: std_logic_vector(7 downto 0);
begin
        g1:xor_gate port map(r(0),t(0),s(0));
        g2:xor_gate port map(r(1),t(1),s(1));
        g3: xor_gate port map(r(2),t(2),s(2));
        g4: xor_gate port map(r(3),t(3),s(3));
        g5: xor_gate port map(r(4),t(4),s(4));
        g6: xor_gate port map(r(5),t(5),s(5));
        g7: xor_gate port map(r(6),t(6),s(6));
        g8: xor_gate port map(r(7),t(7),s(7));
        g9: nor8_gate port map(s,z);
end sgp
```

Schedule of Tasks

Description	Completed	Partially Completed	Not yet started
Design Clock Controller in XILINX	X		
" in LEDIT		X	
" in VHDL			X
Design 16-bit timer in XILINX	X		
" in LEDIT	X		
" in VHDL		X	
Design Register Controller in XILINX	X		
" in LEDIT	X		
" in VHDL	X		
Design Register in XILINX	X		
" in LEDIT	X		
" in VHDL	X		
Design Comparator in XILINX	X		
" in LEDIT	X		
" in VHDL	X		
Combine subsystems in XILINX		X	
" in LEDIT		X	
" in VHDL			X

Questions?