Electrical and Computer Engineering

Digital System Projects Final Practice Problems

By Prawat Nagvajara

Synopsis

Final will cover design with IP cores (Intellectual Property using Vivado IP Catalog and comprehensive knowledge in Digital Design Projects course. Below is a summary.

Base knowledge in digital systems includes the following concepts:

- Digital Systems belong in rational numbers (discrete values and time steps) not in the continuum (real numbers).
- Synchronous blocks – registers , state machine and clock frequency. Sequential circuits for generic radix r adder, serial-adder and pipelined multiplier.
- Combination blocks: electronic switching circuit delay
- Iterative array structures – mapping algorithms to array (synchronous and asynchronous blocks)
- Pipeline structure (pipelining rate and latency)
- Data storages – random access memory, first-in first-out queue and last-in first-out stack
- Test bench
- Fixed-point computational blocks and use of arithmetic packages (std_logic_arith, std_logic_unsigned and std_logic_signed)

The learning mainly achieved through design practices – paper-pencil design, simulation and implementation on FPGA which required skill and knowledge set on IDE tools and FPGA development board technology.

Practice Problems

Problem 1

Design an up/down counter using arith and unsigned package

Problem 2

Design an up/down counter using IP catalog core

Problem 3

Write a test bench for the factorial sequential circuit for n = 1, …, 10.

```
----------------------------------------------------
-- Factorial Sequential Calculator
-- After a reset each clock cycle n,
-- n = 1, 2, ... the output is n!
----------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
entity fact is
generic (K: natural := 32); -- K number of bits
port(ck, reset: in std_logic;
     LED: out std_logic_vector(K-1 downto 0));
end fact;

architecture Behavioral of fact is
signal temp: std_logic_vector(2*K - 1 downto 0);
signal n: std_logic_vector(K-1 downto 0);

begin
process(ck)
begin
if ck='1' and ck'event then
   if reset = '1' then
   -- initialize to vector "00..01"
  temp <= (0=>'1',others=>'0');
  n <= (0=>'1',others=>'0');
  else
  -- K by K product is 2K bits
  temp <= n * temp(K-1 downto 0);
  n <= n+1;
  end if;
end if;
end process;
led <= temp(K-1 downto 0); -- K-bit data
end Behavioral;
```

Problem 4

Design a combinational block to add two N-digit decimal numbers x and y and implement on FPGA for N = 2.

```
entity decimal_N_digit_add is
generic(N: natural := 3);
port (
x, y : in std_loigic_vector(4*N – 1 downto 0);
   z:  out std_loigic_vector(4*N – 1 downto 0));
end decimal_N_digit_add;
```

Problem 5

Design a signed accumulator similar to the IP catalog core with generic parameters, n-bit data and L clock-cycles latency. Use a shift register with depth L-1 to buffer the input data to emulate the latency.