|  |  |
| --- | --- |
|  | **TFE4141 Design of Digital Systems 1** |

**Assignment 1: Terms and definitions used among digital designers**

**Q1: What is 010002 + 010012?**

|  |
| --- |
| 100012 (17) |

**Q2: Which number do 11112 represent?**

1. **In 4-bit unsigned format**
2. **In 4-bit signed format (1’s complement)**

|  |
| --- |
| 1. 15 b) 0 |

**Q2: Which number do 11112 represent?**

1. **In 4-bit unsigned format**
2. **In 4-bit signed format (2’s complement)**

|  |
| --- |
| 1. 15 b) -1 |

**Q3: What does “dynamic range” in the context of number representation mean?**

|  |
| --- |
| Dynamic range is the difference between the largest and smallest values that a number can represent. |

**Q4: What has highest dynamic range of a 32 bit floating point number and a 32 bit fixed point number?**

|  |
| --- |
| The floating point number has a higher dynamic range because the decimal point varies according to its necessity, while the fixed point number cannot move the decimal point (so decimal numbers fixed). |

**Q5: How are floating point numbers and fixed point numbers spaced across the dynamic range?**

|  |
| --- |
| Fixed point numbers are equally spaced across the whole range. The gaps between adjacent values are always the same.  Floating point numbers are unequally spaced between these two extremes, such that the gap between adjacent numbers is much smaller for small values and much larger for large numbers. |

**Q6: Add the numbers from Q1 in the same way as thought at school.**

|  |
| --- |
| 1  01000  + 01001  10001 |

**Q7: A boolean function can be constructed for computing the LSB in the addition of two fixed-point numbers (as done in Q6).**

1. What is a Boolean function?
2. Create a truth table for this Boolean function.
3. Create the Boolean function based on the truth table.

|  |
| --- |
| 1. A Boolean function is a relation between arguments that can only assume 0 or 1. 2. A B | Y Only for the LSB   0 0 | 0  0 1 | 1  1 0 | 1  1 1 | 0 (there should a carry)   1. Y = A  B (or C:\Users\Cristian\Desktop\{displaystyle Acdot {overline {B}}+{overline {A}}cdot B}.png) 🡪 XOR gate |

**Q8: What is a half-adder and what is a full-adder?**

|  |
| --- |
| A **half adder** is a circuit with 2 “1-bit” binary inputs (a & b) and 2 “1-bit” binary outputs (co and s), showing the carry out and sum respectively.  A **full adder** is a circuit with 3 “1-bit” binary inputs (a, b & ci [carry in]) and 2 “1-bit” binary outputs (co and s), also carry out and sum. |

**Q9: Make something cool/useful out of full-adders and half-adders. Draw a diagram**

|  |
| --- |
| http://www.circuitstoday.com/wp-content/uploads/2010/04/Half-Adder-Circuit.gif http://www.circuitstoday.com/wp-content/uploads/2010/04/Full-Adder-Circuit.gif  Half adder (XOR & AND) Full adder (XOR, OR & AND) |

**Q10: Draw symbols for all the logic gates you can think of.**

|  |
| --- |
| NOT:NOT symbol AND:AND symbol OR:OR symbol NAND:NAND symbol  NOR:NOR symbol XOR:XOR symbol XNOR:XNOR symbol |

**Q11: What is sequential logic?**

|  |
| --- |
| **Sequential logic** is a type of logic circuit whose output depends not only on the present value of its input signals, but on the sequence of past inputs, the input history as well (the most basic element in this logic is the flip flop). |

**Q12: What is combinational logic?**

|  |
| --- |
| **Combinational logic** is a type of digital logic which is implemented by Boolean circuits, where the output is a pure function of the present input only (the most basic elements in this logic are the logic gates). |

**Q13: D flip flop**

1. What is a D flip-flop?
2. Draw the symbol commonly used for representing a D flip-flop.
3. Why do some flip-flops have reset?
4. Find VHDL code for a D flip-flop with synchronous reset.

|  |
| --- |
| a) A **D flip-flop** is a circuit that has 2 stable states and can be used to store state information. It captures the value of the D-input at a definite portion of the clock cycle (such as the rising edge).  b) https://upload.wikimedia.org/wikipedia/commons/thumb/8/8c/D-Type_Flip-flop.svg/100px-D-Type_Flip-flop.svg.png  c) They have reset to ensure ‘0’ at start-up (aka synchronization).  d) Library IEEE;  **USE** **IEEE.Std\_logic\_1164.all**;  **entity** **DFlipFlop\_SyncReset** **is**  port(  Q : **out** std\_logic;  Clk : **in** std\_logic;  sync\_reset : **in** std\_logic;  D : **in** std\_logic  );  **end** **DFlipFlop\_SyncReset**;  **architecture** Behavioral **of** **DFlipFlop\_SyncReset** **is**  **begin**  process(Clk)  **begin**  **if** (rising\_edge(Clk)) **then**  **if** (sync\_reset='1') **then**  Q <= '0';  **else**  Q <= D;  **end** **if**;  **end** **if**;  **end** process;  **end** **Behavioral**; |

**Q14: Latch (The evil cousin of the D flip-flop)**

1. What is a latch.
2. Try googling “unwanted latches”.   
     
   Make sure you know everything there is to know about “unwanted latches” before you write any RTL code.

|  |
| --- |
| 1. A latch is the same as a flip flop, but without a clock signal (latches are made by logic gates, and flip flops by latches). 2. Both latches and FF are sequential logic (inside “process”), but the first one is asynchronous and the second one synchronous. As we want all sequential logic to be completely synchronous (in order to avoid undesirable results), one should not define a code inside a process without a “clock signal”, or at least being sure every single output is updated in every clock, whatever condition is given. |

**Q15: What is a register and how are registers related to D flip-flops?**

|  |
| --- |
| A register, made by various D flip flops, updates and holds a value (like the state in a state machine), ideally by using a clock signal. |

**Q16: Draw a 4-bit register**

|  |  |
| --- | --- |
|  | A simple register just takes n inputs in parallel (parallel flip flops), and they are stored a specific quantity of time (determined by the current input and the clock signal). |

**Q17: Draw a 4-bit shift-register**

|  |  |
| --- | --- |
|  | A **shift register** is a cascade of flip flops, sharing the same clock, in which the output of each flip-flop is connected to the 'data' input of the next flip-flop in the chain, shifting by one position the 'bit array' stored in it.  In every clock transition, it 'shifts in' the data present at its input and 'shifts out' the last bit in the array. |

**Q18: What is a Mux?**

|  |
| --- |
| A **multiplexer** (or **mux**) is a device created by logic gates that selects one of several  input signals and forwards the selected input into a single output line. |

**Q19: Google “CPU datapath”, print out one example Datapath and explain what we are looking at.**

|  |  |
| --- | --- |
| http://www.c-jump.com/CIS77/images/datapath.gif | Datapath is the path that the input data follows in a processor to appear as an output. It is made up of the functional units that handle data.  In a microprocessor, the typical units in (at least) one of the datapaths would be: instruction registers (ACC, MAR, MBR), decode latch, ALU registers, load store unit, writeback unit and the memory. |

**Q20: What is the purpose of pipelining in computing?**

|  |
| --- |
| A pipeline is a sequence of modules (or stages) that each performs part of an overall task and passes the partial result to the next stage.  By passing the incomplete task down the pipeline, each stage is able to start work on a new task before waiting for the overall task to be completed. Thus, a pipeline may be able to perform more tasks per unit time than a single module that performs the whole task from start to finish. |

**Q21: What is logic synthesis?**

|  |
| --- |
| Logic synthesis is a process by which an abstract form of desired circuit behaviour, at register transfer level (RTL), is turned into a design implementation in terms of logic gates. The program in charge of that is called synthesis tool. |

**Q22: What is high-level synthesis?**

|  |
| --- |
| Logic synthesis is RTL to Logic Gates, and High level synthesis (HLS) is one level of abstraction above: conversion from a C/C++ program to RTL. With the logic synthesis, the designer provides the RTL description to the synthesizer, and it converts this RTL description to a reliable, optimized gate level representation of the description. HLS converts a C/C++ program to the RTL. It models the transactions happening between the registers, ports, etc. |

**Q23: What does EDA stand for in the context of digital design?**

|  |
| --- |
| Electronic design automation (EDA) is a category of software tools for designing electronic systems such as ICs and PCBs. The tools work together in a design flow that designers use to design and to analyse semiconductor chips. |

**Q24: What does “place & route” refer to?**

|  |
| --- |
| Place & route is a stage in electronic circuits design. It is composed of 2 steps:   * Placement: It decides where to place all electronic components and circuitry in a limited space. * Routing: It decides the exact design of all the wires needed to connect the placed components. |

**Q25: What is static timing analysis (STA)?**

|  |
| --- |
| Static timing analysis (STA) is a simulation method for computing the expected timing of a digital circuit, without requiring a simulation of the full circuit.  The reason behind this is that a complete circuit simulation during the whole design, so as to calculate the delay, is too slow to be practical. |

**Q26: Is negative slack a good thing?**

|  |
| --- |
| **Slack** is the difference between **Required Time RT** (the time a signal needs to arrive at the end of the path to ensure the timing is met) and **Arrival Time AT** (the time that the signal really arrives at the end) for the timing path.  A negative slack is NOT a good thing, because it means that the signal arrives to the end of the path later than it should. |

**Q27: What does “critical path” mean?**

|  |
| --- |
| The critical path is the longest path in the circuit, which really defines the speed of the circuit. |

**Q28: What does the terms “setup-time” and “hold-time” for a flip-flop mean?**

|  |
| --- |
| **Setup Time** is the amount of time the D input must be stable BEFORE the clock edge, so that the data can be stored successfully in the storage device.  **Hold Time** is the amount of time the D input must stay AFTER the clock edge, so that the data can be stored successfully in the storage device.  http://3.bp.blogspot.com/-zV9jxp1bkgc/Uf4FH-S-RgI/AAAAAAAACUk/mHtx9E4PHNs/s1600/setup_hold.jpg |

**Q29: What does “design for testability” (DFT) mean?**

|  |
| --- |
| Design for testability (DFT) consists of IC design techniques that add testability features to a hardware product design. The added features make easier to develop and apply manufacturing tests to the designed hardware, ensuring that the product hardware contains no manufacturing defects that could adversely affect the product's correct functioning. |

**Q30: What is a scan chain?**

|  |
| --- |
| **Scan chain** is a technique used in design for testing every flip flop (by setting and observing) in an IC. It basically shifts-in and shifts-out test data.  It is formed by a number of flip flops connected back to back in a chain with the output of one flop connected to another. |

**Q31: What is formal verification?**

|  |
| --- |
| Formal verification consist of proving the correctness of an electronic circuit with respect to a certain specification, using formal methods of mathematics. |

**Q32: What is a state-machine and what role do they play in digital circuits?**

|  |
| --- |
| State-machine is an abstract mathematical model which has a finite number of inputs, outputs and states. It is used to divide a sequential algorithm in smaller (and independent) sub states in order to make the design easier and more efficient.  State machines are very important to implement all the circuitry with flip flops (sequential logic). |