

EEE4120F Test 2

High Performance Digital Embedded Systems

24 April 2019, 5pm

Duration: 1 hour

Total Marks: 40

MEMO

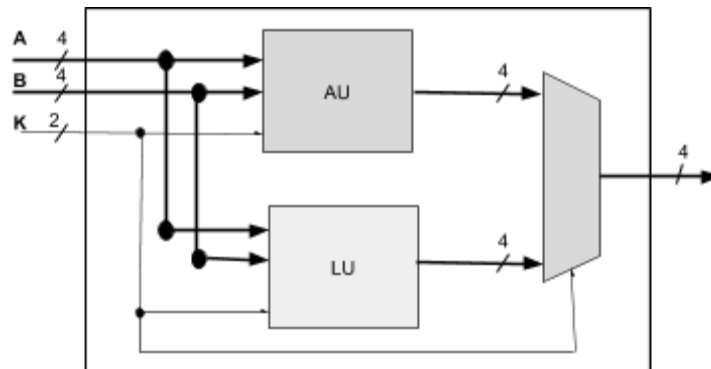
Q1 Explain what FPGA stands for and describe its internal structures [7]

*Field-Programmable Gate Array, ✓ which consists of:
Configurable Logic Blocks ✓ - implement logic functions / using LUTs ✓
Programmable I/O Blocks ✓ - used to interface the outside world / different pin configs ✓
Programmable Interconnects / Switch matrix ✓ - connection/array of wires to connect structures within the FPGA. ✓*

Q2 If clock speeds of CPUs are much faster (>1GHz) than FPGAs (~100MHz), then how/why are FPGAs used to speedup algorithms? [4]

*deterministic - FPGAs have a low jitter, timing is more consistent/concise ✓ ✓
and/or
parallelism - FPGAs can perform many operations at once ✓ ✓
and/or
combinational - certain parts of the algorithm can be implemented with custom circuitry which outperforms CPU operations ✓ ✓*

Q3 You are an engineer tasked with the design of an ALU (Arithmetic Logic Unit) as seen in the diagram below. The arithmetic unit (AU) is already completed, you need to design the logic unit (LU). Please answer the questions that follow.



Q3a Describe using Verilog (at behavioural level) a 4-bit logic unit that implements the following functions, depending on select signal K: [10]

K	Function
00	A and B
01	A or B
10	Complement A
11	A xor B

Sample solution:

```

module logic_unit (A, B, K, F);
    input [1:0] K;
    input [3:0] A, B;
    output reg [3:0] F;
    always@ (A, B, K) begin
        case(K)
            2'b00 : F = A & B;
            2'b01 : F = A | B;
            2'b10 : F = ~A;
            2'b11 : F = A ^ B;
        endcase
    end
endmodule

```

✓✓ module + interface declaration
 ✓✓ port declaration / correct sizes
 ✓✓ selection mechanism on K signal
 ✓ AND operation
 ✓ Or operation
 ✓ COMPLEMENT operation
 ✓ XOR operation

Q3b Provide a line of verilog code to instantiate your module made in Q3a [2]

```

logic_unit mylogicunit (in_A, in_B, in_SEL, out_F); ✓✓
or
logic_unit mylogicunit (.A(in_A), .B(in_B), .K(in_SEL), F.(out_F));

```

Q3c What is the name given to the abstraction level where modules are instantiated and interconnected with signals (as in Q3b) [1]

structural / gate level ✓

Q4 What is quality assurance? [4]

Sample solution:

*Quality is not only about developing a system that **works correctly according to given functional requirements** (✓ for pointing that quality includes meeting functional requirements or about correctness), but which is also: **usable, maintainable, scalable, reliable, reusable, secure and of portable software code** (✓ for one of these desirable non-functional properties).*

*Quality assurance is a **systematic approach** (✓) for ensuring that a **developed system is of good quality** (✓).*

Q5 Poor quality system can contribute to a number of costly HPEC system failures. Read the short narrative below and answer the questions (5a and 5b) that follow.

On June 4, 1996, about 37 seconds after launch, at an altitude of 4Km, the Ariane 5(A5) rocket, carrying a payload of four satellites, deviated 90 degrees from its path due to a software failure, experienced severe aerodynamic stress, and exploded. The A5 program had cost the European Space Agency (ESA) over \$7billion. The success of the earlier Ariane 4 program and budget pressures resulted in the reuse of the A4 software by the A5 program team including its navigation system. An inquiry into the failure found that the Internal Reference System (SRI) that measured the rocket's attitude sent incorrect data to the Flight Control System (FCS) instead of the actual flight data because an arithmetic overflow occurred inside its alignment function (see original code snippet, written in ADA, below) when a 64-bit floating point number for the Horizontal Basis variable (BH) could not be cast and converted to a signed 16-bit integer. The A4 software was re-used as a black box within the A5 and error handling was suppressed for performance reasons.

```
-- Vertical velocity as measured by sensor
L_M_BV_32 := TBD.T_ENTIER_32S ((1.0/C_M_LSB_BV) * G_M_INFO_DERIVE(T_ALG.E_BV));
-- Check, if measured vertical velocity bias can be
-- converted to a 16 bit int. If so, then convert
if L_M_BV_32 > 32767 then
    P_M_DERIVE(T_ALG.E_BV) := 16#7FFF#;
elsif L_M_BV_32 < -32768 then
    P_M_DERIVE(T_ALG.E_BV) := 16#8000#;
else
    P_M_DERIVE(T_ALG.E_BV) := UC_16S_EN_16NS(TDB.T_ENTIER_16S(L_M_BV_32));
end if;
-- Horizontal velocity bias as measured by sensor
-- is converted to a 16 bit int without checking
P_M_DERIVE(T_ALG.E_BH) := UC_16S_EN_16NS (TDB.T_ENTIER_16S ((1.0/C_M_LSB_BH) *
    G_M_INFO_DERIVE(T_ALG.E_BH)));
```

Ariane 4 alignment function ADA Code snippet

Q5a From the given A5 program code team, identify, list and briefly discuss ways in which poor quality system may have led to the A5 disaster. [5]

Sample solution:

- Sacrificing system robustness over performance for mission / safety-critical systems - disabling error handling as a performance optimisation strategy ✓
- Re-using the A4 software as a black box within the A5 - it doesn't seem like the A4 software had even gone through some form of strict certification process to ascertain its quality, it was simply re-used without thorough integration testing because it performed well in the A4 program ✓
- (Any other reasonable points would do)
- The nature of the failure implies that the mission simulation system, if at all present, was either not truly representative of the actual flight - this suggests a limitation in the comprehensiveness of the development environment ✓
- Sacrificing quality in order to commission a mission / safety critical system on time and economically - from the story, it seems the A5 program had tight budgetary constraints which subtly influenced them to opt for easier and quicker design decisions but which were eventually very costly ✓
- Insufficient component and system testing - insufficient testing for components reused from Ariane 4 were the cause of the failure ✓

Q5b Suppose that you are part of a design team that is working on A5++, a new version of A5 with an improved SRI system. Discuss how you would incorporate quality into the subsystem in order to minimize the occurrence of similar and other disasters that can arise due to poor quality system. To be more specific, describe how you would address the quality limitations identified in the narrative above. Make reference to the code snippet in your description and show or explain how it could be modified. [7]

Sample solution:

- Use **formal specifications** for system modeling, analysis, verification and validation - formal specification suitable for mission and safety-critical systems as they help achieve reliability and predictability ✓ ✓
- Specifically, use **pre and post-conditions** to ensure that system methods operate on valid input data and in turn generate valid output data ranges ✓
- Ensure that the last statement has its **boundary values checked** ✓, to avoid an **overflow exception being thrown** ✓. A proper replace for that statement could be: ✓ ✓ ✓ ✓ ✓

```
L_M_BH_32 := TBD.T_ENTIER_32S ((1.0/C_M_LSB_BH) * G_M_INFO_DERIVE(T_ALG.E_BH));
if L_M_BH_32 > 32767 then
  P_M_DERIVE(T_ALG.E_BH) := 16#7FFF#;
elsif L_M_BH_32 < -32768 then
  P_M_DERIVE(T_ALG.E_BH) := 16#8000#;
else
  P_M_DERIVE(T_ALG.E_BH) := UC_16S_EN_16NS(TDB.T_ENTIER_16S(L_M_BH_32));
end if;
```