EEL 4712
Midterm 3 – Spring 2016
VERSION 1

Name: _____

UFID: _____

Sign here to give permission for your test to be returned in class, where others might see your score:

_____

IMPORTANT:
• Please be neat and write (or draw) carefully. If we cannot read it with a reasonable effort, it is assumed wrong.
• **As always, the best answer gets the most points.**

COVER SHEET:

Problem#:	Points
1 (5 points)	
2 (5 points)	
3 (5 points)	
4 (5 points)	
5 (5 points)	
6 (5 points)	
7 (5 points)	
8 (5 points)	
9 (5 points)	
10 (14 points)	
11 (5 points)	
12 (6 points)	
13 (5 points)	
14 (5 points)	
15 (20 points)	

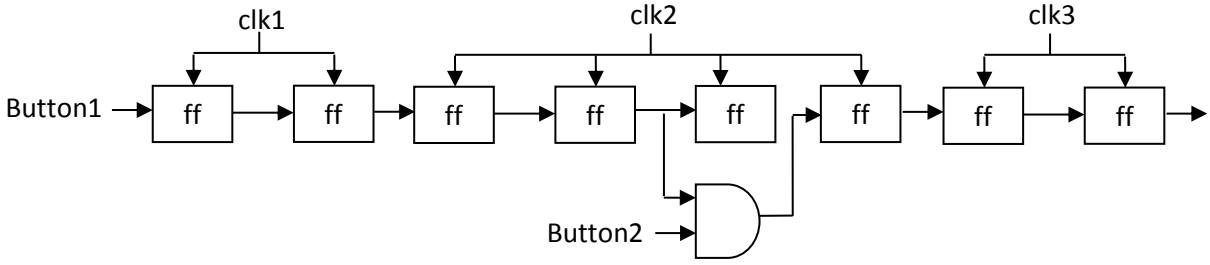
Total:

Regrade Info:

1) (5 points) Show the synthesized FPGA circuit for a VHDL architecture that uses tristates to allow 8 potential sources for a single wire.

2) (5 points) Metastability is generally unavoidable in which of the following situations (circle all that apply).
   a. Clock-domain crossing
   b. Finite state machines
   c. Asynchronous inputs to a register
   d. Communication within a single clock domain
   e. Combinational logic
   f. Technology mapping

3) (5 points) At what points in the following circuit will metastability eventually occur?



4) a. (2.5 points) What synchronizer should be used to synchronize multiple bits across clock domains when throughput **is not** important?

   b. (2.5 points) What synchronizer should be used to synchronize multiple bits across clock domains when throughput **is** important?

5) (5 points) To perform a subtract without a borrow, what instruction must the assembly code include right before the SBCR instruction?

6) (5 points) For load instructions using absolute addressing, where is the data stored that will be loaded into the accumulator register?

7) (5 points) For load/store instructions using immediate addressing, where is the data stored that will be loaded into the accumulator register?

8) (5 points) For a SP that is currently set to 0x0100, what would be the SP after three call instructions (assume there are no return instructions in between the calls).

9) (5 points) Consider the situation where we have added a timer peripheral to the small8 external bus that is memory mapped to address 0x1000. This timer provides 8-bit values that specify how many seconds have elapsed since the last time the peripheral was read. Write an assembly program using the small8 instruction set that reads from this peripheral and outputs the value to outport 0. Use whatever assembly syntax you want.

10) (14 points) Create a memory initialization file for the following assembly code. Add a comment to show the beginning of each instruction and each variable in memory. *Break your answer up into two columns and/or use the following page.*

```
INPORT0         EQU     $FFFE
INPORT1         EQU     $FFFF
OUTPORT0        EQU     $FFFE

BEGIN:
        LDAA    INPORT0
        STAR    D
        LDAA    INPORT1
        ANDR    D
        LDAA    MASK
        ANDR    D
        BEQA    SKIP
        SBCR    D
SKIP:
        ADCR    D
        STAA    OUTPORT0

LOOP:
        SETC
        BCSA    LOOP

* Data Area
MASK:   dc.b    $55

        END     BEGIN
```

```
Depth = 256;
Width = 8;
Address_radix = hex;
Data_radix = hex;
% Program RAM Data %
Content
  Begin
```

```
[      ..00FF] : 00;
End;
```

11) (5 points) Describe what happens when the following VHDL code is simulated:

```
process(count)
begin

     count <= count + 1;

end process;
```

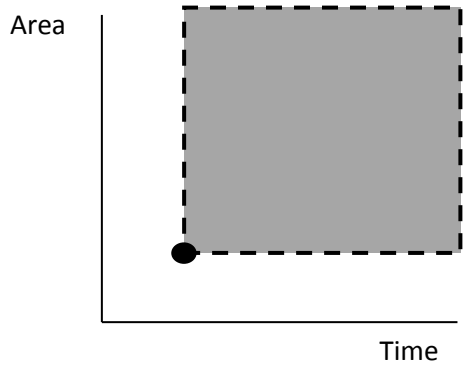
12) a. (3 points) Create a constrained array type in VHDL with 100 elements, where each element is 32 bits.

b. (3 points) Create an unconstrained array type in VHDL where each element is 16 bits. Instantiate a signal of this type with 500 elements.

13) (5 points) Given a solution space with the following implementations, which of the solutions are ***not*** Pareto optimal? If they are all Pareto optimal, state that.

a. Area: 1000 LUTs, Time: 8s
b. Area: 2000 LUTs, Time: 9s
c. Area: 3000 LUTs, Time: 8.5s
d. Area: 4000 LUTs, Time: 6s
e. Area: 5000 LUTs, Time: 4s

14) (5 points) **True or false**: the single point shown in the design space provides enough information to conclude that any solution in the gray region is not Pareto optimal. Explain your answer if necessary.



15) a. (8 points) For the following code, create a schedule for the provided datapath. Ignore muxes, registers, and other glue logic. Like the examples in class, assume that address calculations are done *without* using the specified resources (i.e., address calculations cost nothing). Do not change the code. List any assumptions.

```
for (int i=0; i < 1000000; i++) {
   a[i] = b[i]*411 + c[i]*215 + d[i]*24 + e[i]*17;
}
```

Datapath
4 multipliers
2 adders
1 comparator
1 memory for b[] (can read 4 elements/cycle)
1 memory for a[] (can write 1 element/cycle)

b. (4 points) What is the execution time in total cycles based on your schedule from part a? Show your work.

c. (4 points) For a pipelined implementation of this datapath, what is the approximate execution time in total cycles?

d. (4 points) What is the name of the optimization that performs multiple iterations of a loop at the same time?

INSTRUCTION	OP CODE	C	V	Z	S	DESCRIPTION	SYNTAX
Load Acc (Imm)	84	×	×	✓	✓	A ← mem[PC]	LDAI <data>
Load Acc (Abs)	88	×	×	✓	✓	A ← mem(mem[PC])	LDAA <address>
Load Acc (RR)	81	×	×	✓	✓	A ← (D)	LDAD
Store Acc (Abs)	F6	×	×	×	×	Mem(mem[PC]) ← (A)	STAA <address>
Store Acc (RR)	F1	×	×	×	×	D ← (A)	STAR D
Add with Carry	01	✓	✓	✓	✓	A ← (A) + (D) + C	ADCR D
Subtract with Borrow	11	✓	✓	✓	✓	A ← (A) + not(D) + C	SBCR D
Compare	91	✓	✓	✓	✓	Same as Subtract, but only change Status Flags (A is unchanged)	CMPR D
AND	21	×	×	✓	✓	A ← (A) AND (D)	ANDR D
OR	31	×	×	✓	✓	A ← (A) OR (D)	ORR D
XOR	41	×	×	✓	✓	A ← (A) XOR (D)	XORR D
Shift Left Logical	51	✓	×	✓	✓	C ← (A7), A7 ← A6, …, A0 ← 0	SLRL
Shift Right Logical	61	✓	×	✓	✓	C ← (A0), A0 ← A1, …, A7 ← 0	SRRL
Rotate Left through Carry	52	✓	×	✓	✓	C ← (A7), A7 ← A6, …, A0 ← C	ROLC
Rotate Right through Carry	62	✓	×	✓	✓	C ← (A0), A0 ← A1, …, A7 ← C	RORC
Branch on /C (Inh)	B0	×	×	×	×	if (C=0), PC ← mem[PC] else PC++	BCCA
Branch on C (Inh)	B1	×	×	×	×	if (C=1), PC ← mem[PC] else PC++	BCSA
Branch on Z (Inh)	B2	×	×	×	×	if (Z=1), PC ← mem[PC] else PC++	BEQA
Branch on S (Inh)	B3	×	×	×	×	if (S=1), PC ← mem[PC] else PC++	BMIA
Branch on /Z (Inh)	B4	×	×	×	×	if (Z=0), PC ← mem[PC] else PC++	BNEA
Branch on /S (Inh)	B5	×	×	×	×	if (S=0), PC ← mem[PC] else PC++	BPLA
Branch on /V (Inh)	B6	×	×	×	×	if (V=0), PC ← mem[PC] else PC++	BVCA
Branch on V (Inh)	B7	×	×	×	×	if (V=1), PC ← mem[PC] else PC++	BVSA
Decrement Acc	FB	×	×	✓	✓	A ← (A) - 1	DECA
Increment Acc	FA	×	×	✓	✓	A ← (A) + 1	INCA
Set Carry Flag	F8	✓	×	×	×	C ← 1	SETC
Clear Carry Flag	F9	✓	×	×	×	C ← 0	CLRC

Table 1: Small8 Instruction Set