ECSE 485 **Lab 3** **Due: Oct 20**

In this lab, you are asked to design the ALU of a 16-bit RISC microprocessor. You can find example in ‘‘Computer architecture: a quantitative approach’’ (by David A Patterson and John L. Hennessy). This book introduces a famous RISC example, DLX. You can look at other references if you wish, but you need to follow the specification given below.

Develop a hierarchical design by first identifying the modules which will implement the specification. Refine the design, till the lowest level cells contain transistors and gates.

Top cell : alu

I/O: bit 15 is the most significant bit. INPUTS: A(15:0) , B(15:0) , alu_code(4:0) ; OUTPUTS: C(15:0) overflow

The ALU control signals (alu_code) consist of five bits and define a total of 21 instructions.

- There are two fields in these five signals: operation type field, and operation field.
- The two high order bits denote the operation type and the lower three bits represent the operations which will take place.
- The bit patterns for the operation type field are:

arithmetic operation (00---)			**logic operations (01---)**		
add : 000	signed addition	(A+B -> C)	and : 000	A AND B	(A AND B -> C)
addu : 001	unsigned addition	(A+B -> C)	or : 001	A OR B	(A OR B -> C)
sub : 010	signed subtraction	(A-B -> C)	xor : 010	A XOR B	(A XOR B -> C)
subu : 011	unsigned subtraction	(A-B -> C)	not : 100	NOT A	(NOT A -> C)
inc : 100	signed increment	(A+1 -> C)			
dec : 101	signed decrement	(A-1 -> C)			

shift operations (10---)		**set condition operation (11---) (A and B are signed numbers)**		
sll : 000	logic left shift A by the amount of B	sle : 000	if A <= B then	C(15:0) = <0...0001>
srl : 001	logic right shift A by the amount of B	slt : 001	if A < B then	C(15:0) = <0...0001>
sla : 010	arithmetic left shift A by the amount of B	sge : 010	if A >= B then	C(15:0) = <0...0001>
sra : 011	arithmetic right shift A by the amount of B	sgt : 011	if A > B then	C(15:0) = <0...0001>
		seq : 100	if A = B then	C(15:0) = <0...0001>
		sne : 101	if A != B then	C(15:0) = <0...0001>
			otherwise, set	C(15:0) = <0...0000>

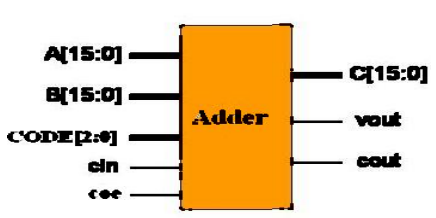
- If there is an overflow in the ALU operations, the signal ``overflow" should be set to high, otherwise it is low. Therefore, when doing all the operations other than arithmetic ones, ``overflow" should always be set to low.
- Because this is a 16-bit ALU, the amount of shift will never go over 15. Only B (3:0) are used to define the shift amount.
- Find the critical path of your ALU. Try to optimize the circuit on the critical path so the delay of your ALU can be minimized.

TOP MODULE: 16-bit Adder Module

PRIMARY SIGNALS: BUS inputs/outputs and additional signals

BUS signals: bit 15 is the most significant bit. All bus signals are named with upper case letters

INPUTS: A[15:0], B[15:0],CODE[2:0] OUTPUTS: C[15:0]

Additional signals: named with lower case letters	**3-bit CODE[2:0] control signals**	
vout = signed overflow cout = carry output cin = carry input coe = carry output enable (active low)	add : 000 signed addition (A+B -> C) addu : 001 unsigned addition (A+B -> C) sub : 010 signed subtraction (A-B -> C) subu : 011 unsigned subtraction (A-B -> C) inc : 100 signed increment (A+1 -> C) dec : 101 signed decrement (A-1 -> C)	

References

1. Read chapter 11 of the textbook.

Submitting and Grading

Submit the following:

- One hard copy of your schematic design, including all levels of your hierarchy.
- One hard copy of the result waveform (make sure to explain your waveforms)
- Briefly illustrate (one paragraph is enough) on what the critical path is and explicitly give the worst case delay.
- Save your final version of design. You may be asked to give demo and the submit date will, in this case, is based on the date you saved your final version.

Grading policy:

- Correctness of design (70%)
- Performance (delay of worst case path(s)) (25%)
- Organization of your design (5%)
- Top-down design, separation of blocks, number of gates used, ...
- Penalty for late submission: 5% per working day (maximum 25%)
- Bonus for early submission: 5% per working day (maximum 10%).

NOTICES for the Assignment

- You are allowed to use only "Basic Logic Gates" and "transistors" in ADK IC library for your ALU design.
- Generate proper inputs to verify your ALU functions.
- Generate inputs that exercise the identified critical path.