

**Midterm Exam**  
**15 points total**

**March 28, 2011**

**Part I**  
**Analytical Problems**

1. **(1.5 points)**

A. Convert to decimal, compare, and arrange in ascending order the following numbers encoded using various binary signed number representations:

- a.  $11011011.1101_{SM}$
- b.  $01011011.1101_{OC}$
- c.  $11011011.1101_{TC}$
- d.  $01011011.1101_B$

Notation: SM - signed magnitude representation,  
OC - one's complement representation,  
TC - two's complement representation,  
B - biased representation with the bias  $B=2^7 = 128$ .

B. Extend all of the above representations to the equivalent representations with 12 bits in the integer part and 8 bits in the fractional part.

2. **(1.5 points)** Determine all bits of the ANSI/IEEE standard single-precision representation of the following numbers:

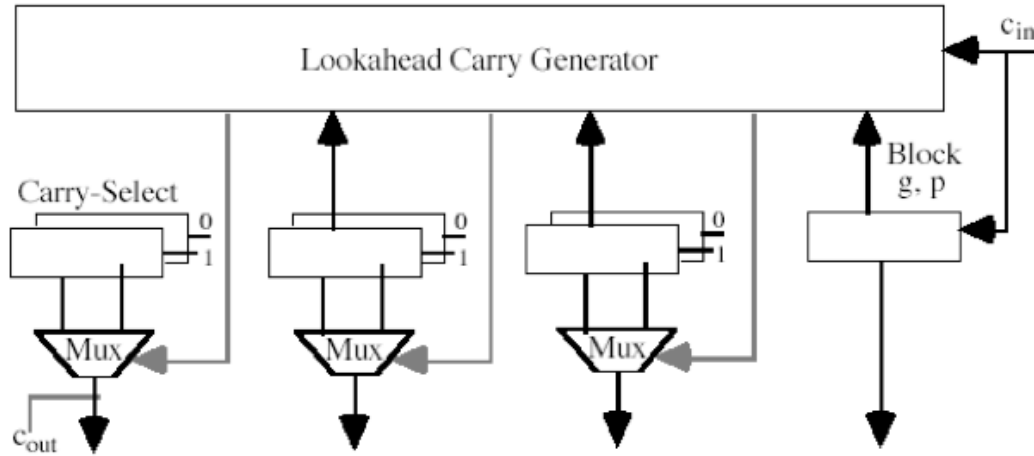
- a.  $-4.DDDDDD_{16} \times 16^{-32}$
- b.  $ABC.DDDDDD_{16} \times 16^{-34}$
- c. smallest positive ordinary number + largest positive ordinary number
- d. smallest positive denormal  $\times$  largest positive ordinary number

**Hint:** Use the default IEEE rounding scheme whenever appropriate.

3. (1.5 points) Arrange the listed below signals in the order they are generated within a 16-bit Hybrid Carry-Lookahead/Carry-Select Adder, shown in the diagram below: For each signal calculate the time when this signal becomes stable for arbitrary values of inputs, assuming that all inputs change at the time 0.

Assume that

- The adder is built out of AND, OR, and XOR gates with up to 5 inputs, and the delays of all these gates are equal.



- A.  $s_7$
- B.  $p_{[11, 8]}$
- C.  $c_{out}$
- D.  $c_8$
- E.  $s_{10}$
- F.  $g_{[7, 4]}$

**Part II**  
**Design Problems**  
**(3.5 points for each problem)**

1. Draw full block diagrams of
  - a. 8-bit Hybrid Brent-Kung/Kogge-Stone Parallel Prefix Network Adder, and
  - b. 8-bit Conditional Sum Adder.

Using these diagrams, show all intermediate values in each node of each adder when performing the operation  $X + Y + c_0$  for the following values of inputs:

$$X = 10110110$$

$$Y = 11001011$$

$$c_0 = 1$$

Verify that the obtained result, including  $c_{out} = c_8$ , is correct and the same for both adders.

2. Design and compare the following two circuits:
  - a. k-bit ripple-carry incrementer (RCI), and
  - b. k-bit digit-serial incrementer (DSI) with the digit size d.

Assume that

- all inputs, including control inputs, are registered (outside of your circuit).
- delays of all gates are independent of the number of inputs, and are equal to the delay of an inverter. Thus, you can assume that the delay of each gate and inverter is equal to 1.
- area of each gate is proportional to the number of inputs. For example, area of a 2-input gate is equal to the area of 2 inverters. Assume that the area of an inverter is equal to 1, area of a 2-input gate is equal to 2, etc.
- area of a D flip-flop is equal to 12, its clock-to-output delay is equal to 2, and its setup time is equal to 1.
- do not include the areas of the surrounding input and output registers in your computations.

Perform the following tasks:

- a. Draw a block diagram of each circuit using medium level components, such as multiplexers, half-adders, modified half-adders, full-adders, flip-flops, etc.
- b. Draw a detailed schematic of all combinational components of your circuits (no schematic required for a flip-flop).
- c. Determine areas and delays of all combinational components of your circuit.
- d. Determine general formulae for the Clock Period (T), Latency (L), and Area (A) of each incrementer as a function of k and d.
- e. Derive formulae for the following ratios
  - $L_{DSI}(k,d)/L_{RCI}(k)$
  - $A_{RCI}(k)/A_{DSI}(k,d)$
  - $L_{RCI}(k)*A_{RCI}(k)/L_{DSI}(k,d)*A_{DSI}(k,d)$ .
- f. Calculate values of the above ratios for  $k=32, 64, 128$  and  $d=8$ .

**Warning: Please note that using corresponding adders (i.e., ripple carry adder and digit serial adder) instead of incrementers will be treated as a highly inefficient solution, and may result in receiving zero points for this problem.**

3. Design a circuit that calculates an average of 16 unsigned 4-bit numbers.

The circuit accepts four operands (V, X, Y, Z) in the first clock cycle, and three operands (X, Y, Z) in the following four clock cycles.

This averaging circuit is specified below using its pseudocode and block diagram.

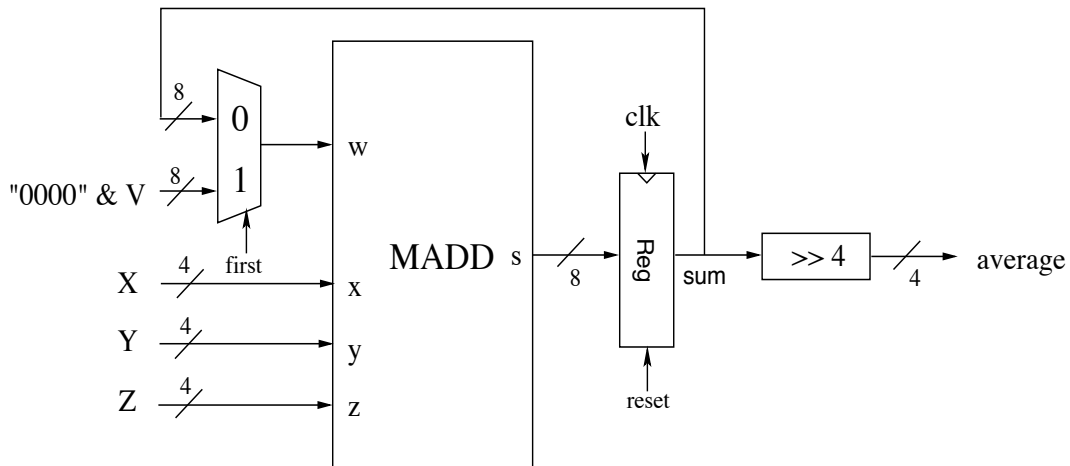
### Pseudocode

```

sum = 0
for i = 0 to 4 do
  if i = 0 then
    sum = V + X + Y + Z
  else
    sum = sum + X + Y + Z
  end if
end for
average = sum / 24

```

### Top-level Block Diagram



Design and analyze the multi-operand adder MADD being a part of this circuit.

In order to do that, please perform the following tasks:

- a. Draw the dot diagram of the operations performed by MADD.
- b. Draw a block diagram of MADD using medium level components, such as full-adders and half-adders.
- c. Clearly mark names and indices of all signals in your schematic, using notation such as  $x_0$ ,  $w_1$ ,  $s_{10}$ ,  $c_{13}$ , etc.
- d. Determine the minimum number of 4-input Look-Up Tables (LUTs) necessary to implement this circuit using Xilinx Spartan 3 devices, assuming that
  - Full adders within Carry Save Adders are implemented using LUTs only.
  - Full adders within a Carry Propagate Adder and half-adders within both types of adders are implemented using fast carry chain logic (LUT + 2-to-1 MUX + XOR gate).

**Bonus (1 point):**

Determine the critical path and delay of this circuit assuming that

- the delay of a LUT is 1 ns, the delay of a fast-carry-chain MUX is 0.1 ns from carry in to carry out, and 0.25 ns from the select signal to carry out, and the delay of a fast-carry-chain XOR gate is 0.25 ns
- delays of interconnects should be neglected in your computations.