EE216B Practice Midterm – Solution

PROBLEM 1: DSP Arithmetic

Assume you have a full adder block with input bits a, b, C_{in} and outputs S_{out} and C_{out}.

a. Write the logic expressions for S_{out} (sum) and C_{out} (carry out) in terms of the 1-bit input operands a, b, and C_{in} (carry in).

$S_{out} = a \oplus b \oplus C_{in}$

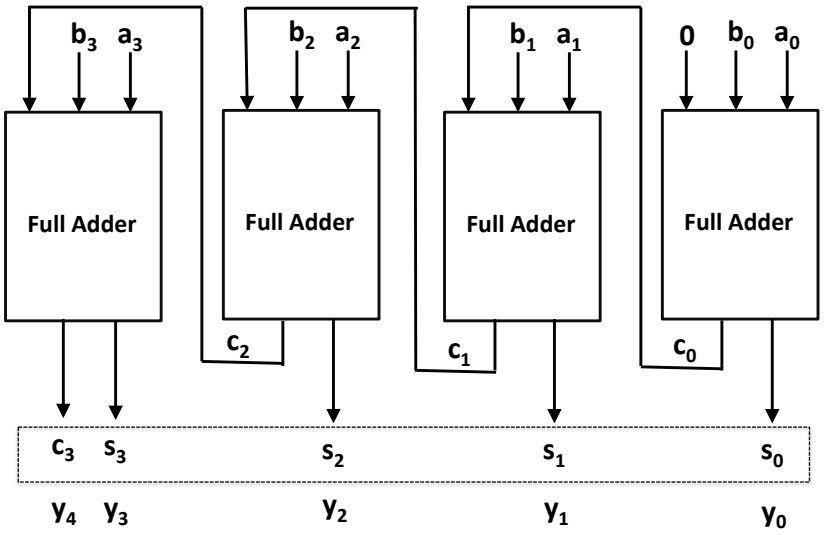
$C_{out} = a.b + C_{in}(a+b)$

b. Use this full adder to design a system that performs the following operation:

Y = A + B

Here $A = a_3a_2a_1a_0$ and $B = b_3b_2b_1b_0$ are 4-bit unsigned numbers. Clearly indicate the inputs to all the full adder blocks used in your system. Also clearly mark the output Y of your system.

Y is a 5-bit binary number. For the unsigned case Y is computed as:

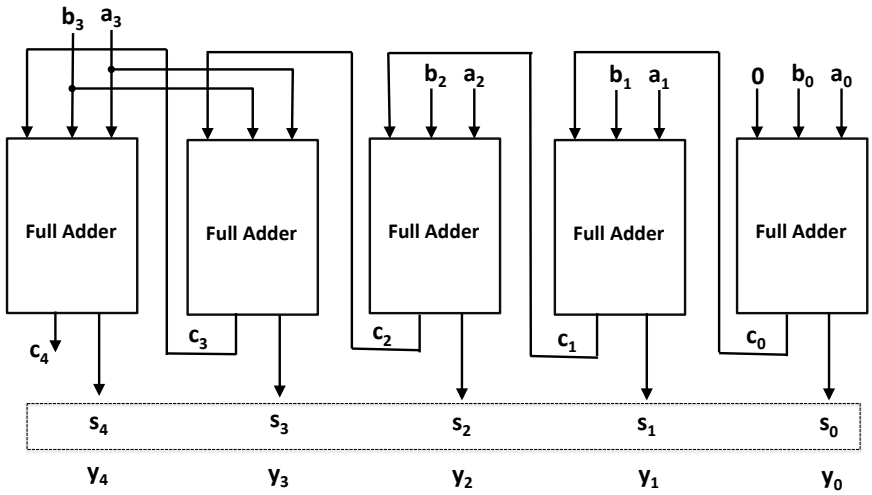


c. What is the output Y of the system designed in (b) when:

i) A = 1001
B = 1100
Y = 10101

ii) A = 1100
B = 0110
Y = 10010

d. Redesign the system in (b) when A and B are in 2's complement representation. Again clearly indicate the inputs to all the full adder blocks and also clearly mark the output Y of your system.



e. What is the output Y of the system designed in (d) when:

i) A = 1001
B = 1100
Y = 10101

ii) A = 1100
B = 0110
Y = 00010

PROBLEM 2: Architectural Evaluation

Background: Architecture efficiency can be simply evaluated as ε = Performance/Cost. Synchronously clocked implementations are convenient for the study of efficiency, because the clock period T_{CLK} is a reference for evaluating both the performance and throughput. The performance in terms of the computational rate is given by $R_C = N_{op}/T_{CLK}$ (N_{op} = number of operations) and the throughput is given by $R_T = N_S/T_{CLK}$ (N_S = number of samples). Due to the common reference clock, computational rate and throughput are proportional $R_C = (N_{op}/N_s)R_T$. The relationship between (throughput, performance) and chip size represents a measure of the efficiency of an architecture.

Goal: evaluate the efficiency of pipelined and parallel architectures, relative to the baseline architecture. Assume following models, where N indicates the degree of parallelism or pipelining:

Parameter / Model	Baseline	Parallel	Pipeline
Throughput	R_1	$N \cdot R_1$	$R_1 \frac{N}{1+(N-1)\cdot\frac{T_{REG}}{T_1}}$
Area	A_1	$N \cdot A_1 + (N-1)\cdot A_Z$	$A_1 + (N-1)\cdot A_{REG}$

The throughput for the pipelined architecture is derived assuming that the logic is pipelined such that each datapath unit has the same delay. T_{REG} is the delay of the pipeline register. A_Z is the additional area for data distribution and data merging in parallel realization.

a. Assuming that the efficiency of the baseline architecture is $\varepsilon_1 = R_1/A_1$, calculate the efficiency of parallel architecture relative to ε_1. Find N_{opt} that that maximizes the efficiency for $A_Z/A_1 = 0.1$. Explain your result.

$$\varepsilon_N = \frac{N R_1}{A_1\left[N + (N-1)\frac{A_Z}{A_1}\right]} = \varepsilon_1 \frac{1}{1 + \frac{N-1}{N}\frac{A_Z}{A_1}}$$

$\varepsilon_N = \max$ for $\boxed{N_{opt} = 1}$ $\quad \varepsilon_{N_{opt}} = \varepsilon_1$

THROUGHPUT ↑ BY N
AREA ↑ BY $> N$
$\Rightarrow \varepsilon_N \leq \varepsilon_1$

b. Assuming that the efficiency of the baseline architecture is $\varepsilon_1 = R_1/A_1$, calculate the efficiency of pipelined architecture relative to ε_1. Find N_{opt} that that maximizes the efficiency for $A_{REG}/A_1 = 0.1$ and $T_{REG}/T_1 = 0.1$. Explain your result.

$$\mathcal{E}_N = \frac{N R_1}{1 + (N-1)\frac{T_{REG}}{T_1}} \cdot \frac{1}{A_1\left[1 + (N-1)\frac{A_{REG}}{A_1}\right]}$$

$$= \mathcal{E}_1 \frac{N}{\left[1 + (N-1)\frac{T_{REG}}{T_1}\right]\left[1 + (N-1)\frac{A_{REG}}{A_1}\right]}$$

$$= \mathcal{E}_1 \frac{N}{\left[1 + \frac{N-1}{10}\right]^2} \Rightarrow \frac{\partial \mathcal{E}_N}{\partial N} = 0 \Rightarrow \boxed{N_{opt} = 9}$$

$$\mathcal{E}(N_{opt}) = 2.77\,\mathcal{E}_1$$

Since a pipeline register has a smaller contribution on area and delay than datapath logic,

$N_{opt} > 1$ and $\varepsilon(N_{opt}) > \varepsilon 1$

PROBLEM 3: FIR Filter Architecture

a. Consider "folded array" filter architecture for MIMO systems as shown below.

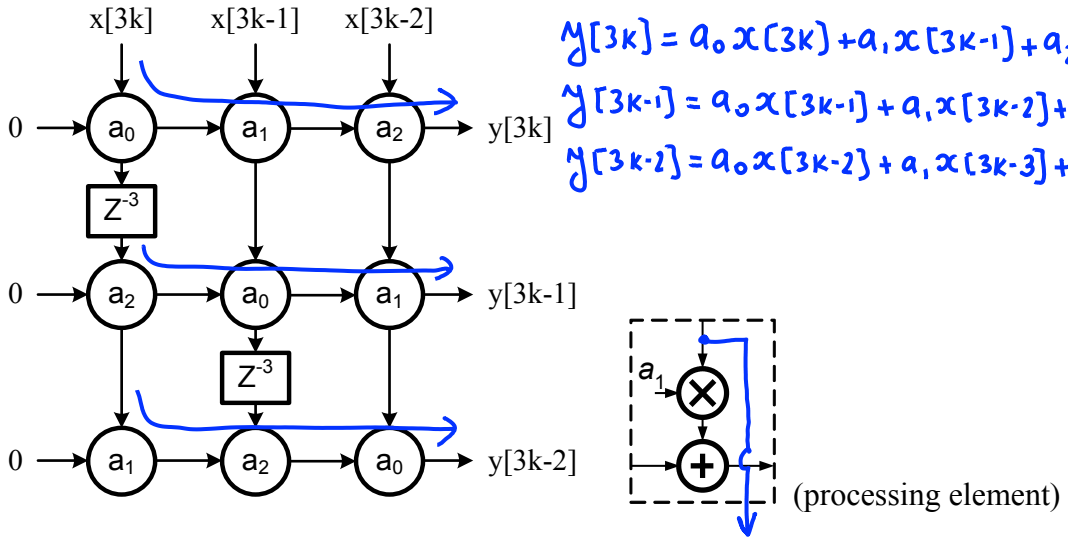
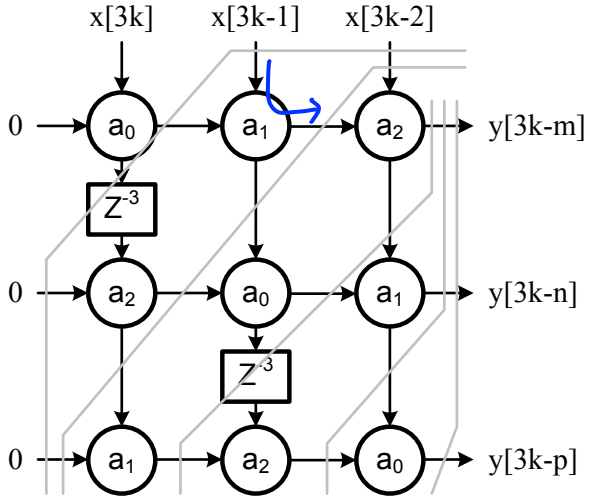


Figure 1: Folded array implementation of a 3-tap FIR filter for MIMO systems.

What is the critical path of this architecture? Assume t_{adder} = 200ps, $t_{multiplier}$ = 1ns. Highlight the critical path in the figure above. How many paths are critical?

$t_{crit-path} = t_{mult} + 3t_{add} = 1.6ns$

b. To further improve the throughput, word-level pipelining is performed as shown below (gray lines indicate pipeline cut-sets). What are the resulting latencies *m*, *n*, and *p* of the output samples? How much did the throughput improve?



$t_{crit-path} = t_{mult} + t_{add} = 1.2ns$

$\frac{1.6}{1.2} = 1.33X$ THROUGHPUT IMPROVEMENT

$m = 5$
$n = 6$
$p = 7$