

ECE 645
Spring 2014

Homework 3 Bonus

due Saturday, March 1, 11:59 PM

Instructions:

Solve one of the following two bonus problems:

Analytical Problem *or* Implementation Problem.

If you solve one of these problems, you may still obtain up to 50% of points for solving the other problem.

The Implementation Problem requires knowledge of:

- VHDL *or* Verilog, and
- Xilinx FPGA tools.

Common Background:

The operation of a high-radix parallel prefix network adder is shown in Figs. 1-4.

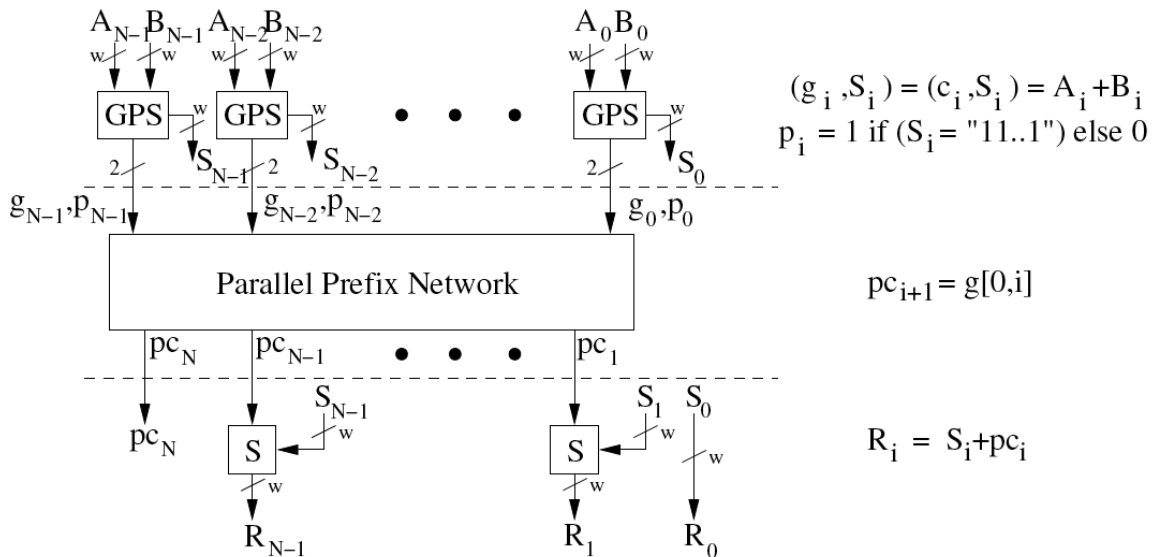


Fig. 1 Block diagram of a high-radix parallel prefix network adder.

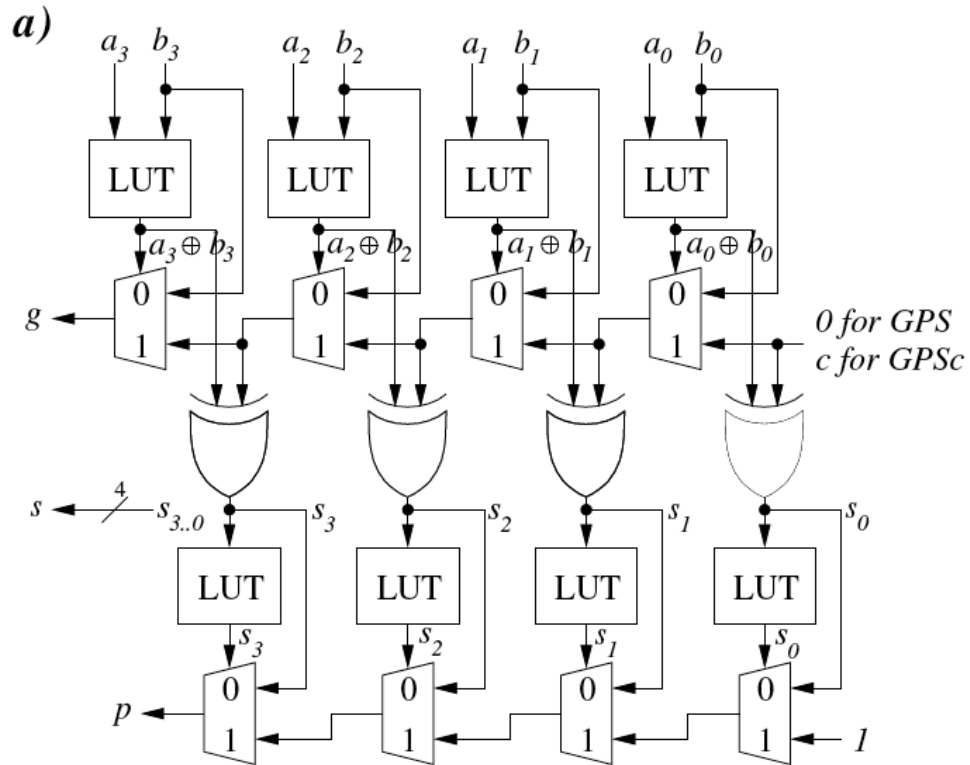


Fig. 2 Block diagram of the Generate-Propagate-Sum (GPS) unit for $w=4$, implemented using internal resources of Xilinx FPGAs.

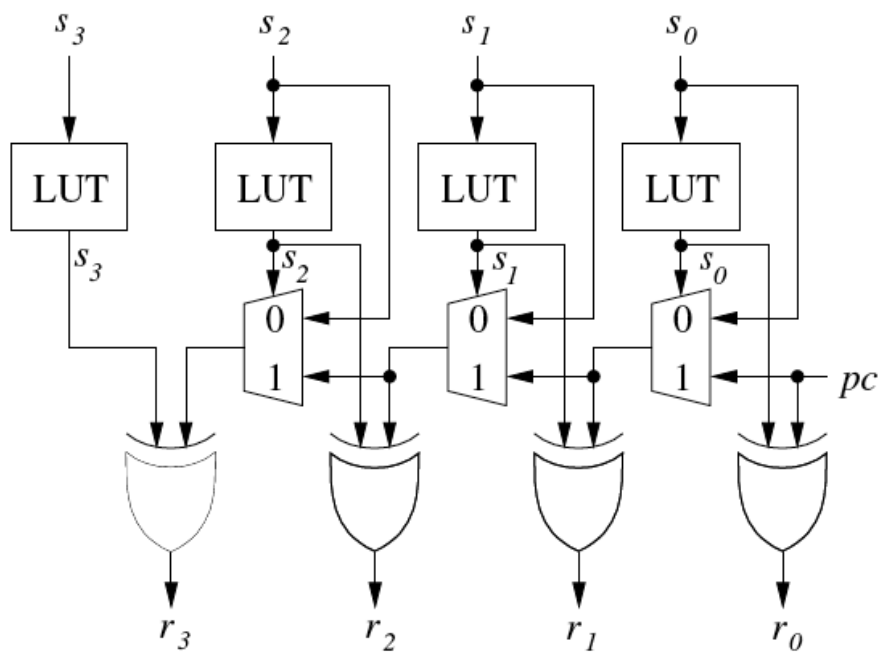
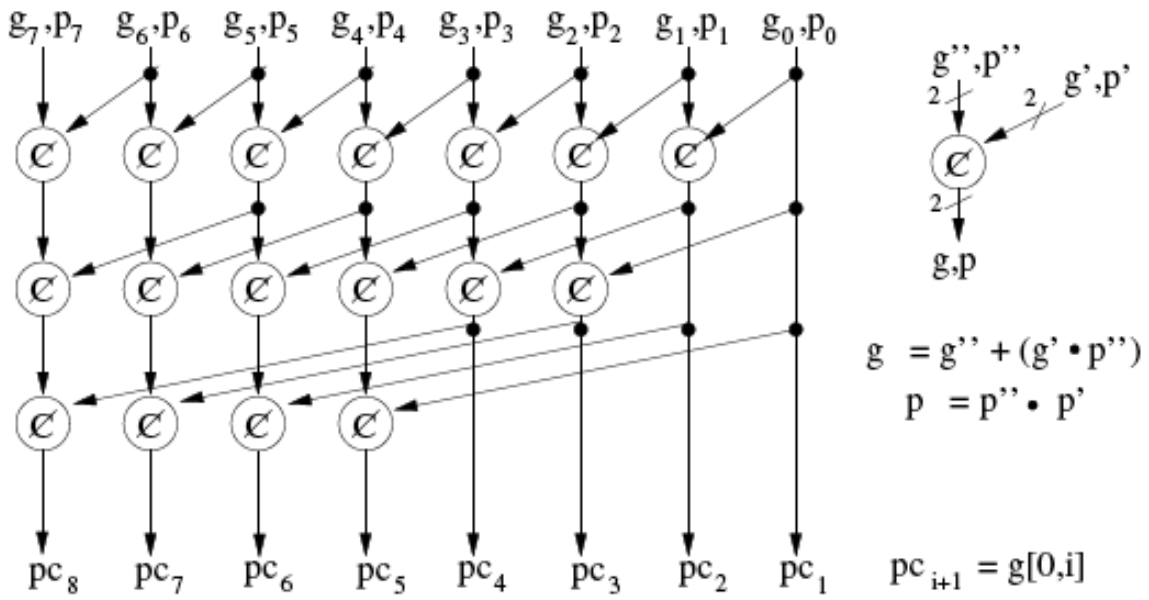


Fig. 3 Block diagram of the sum unit (S) for $w=4$, implemented using internal resources of Xilinx FPGAs.

a)



b)

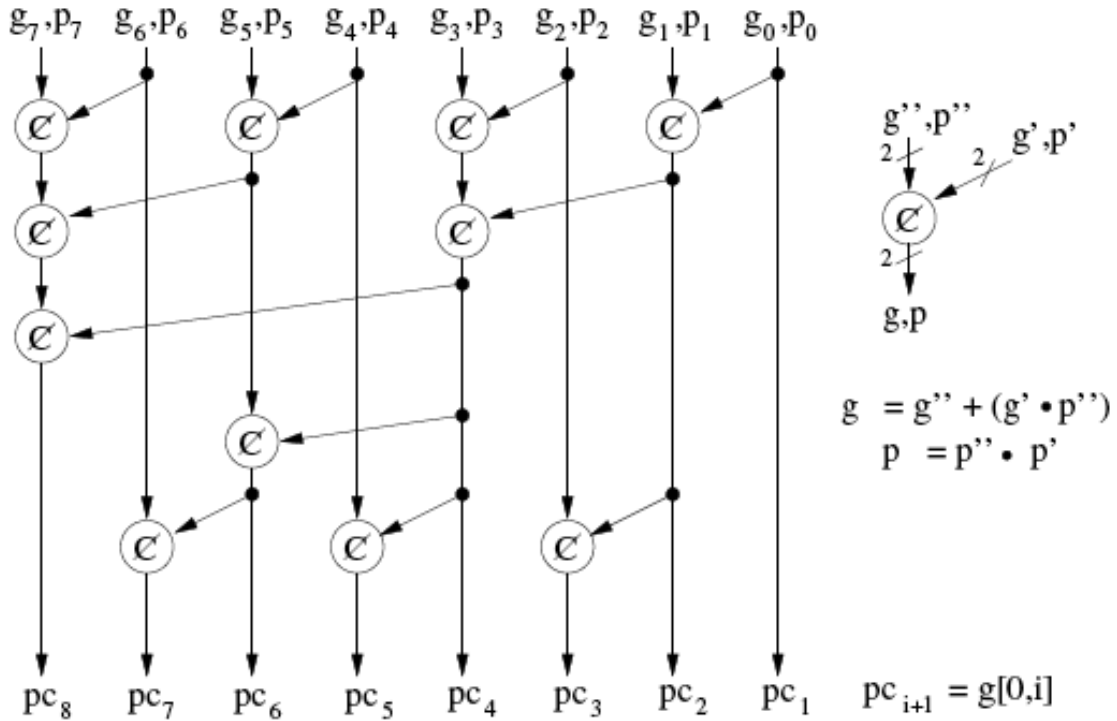


Fig. 4 Parallel Prefix Networks for N=8, a) Kogge-Stone, b) Brent-Kung.

Analytical Problem

Assumptions:

Assume the following values of individual delays for an FPGA used:

$$d_{\text{LUT}} = 0.75 \text{ ns}$$

$$d_{\text{XOR2}} = 0.20 \text{ ns} \quad (\text{applies only to the hardwired XORs shown in Figs. 2 and 3})$$

$$d_{\text{MUX2}} = 0.075 \text{ ns} \quad (\text{applies only to hardwired multiplexers, shown in Figs. 2 and 3})$$

$$d_{\text{FF}} = 0.20 \text{ ns}$$

$$t_{\text{setup}} = 0.20 \text{ ns}$$

In order to exit the carry and control logic, c_{out} of each Ripple-Carry Adder, must pass as c_{in} through the hardwired XOR2 of the next adder stage.

Delays of all interconnects are assumed to be equal to zero.

Task 1

Derive the formulas for the latency of the High-Radix Parallel Prefix Network Adder for

- A. Kogge-Stone Parallel Prefix Network
- B. Brent-Kung Parallel Prefix Network

as a function of the operand size $n=2^k$ and the word size w . Note that $N=n/w$.

Assume that the adder is surrounded by registers, and include the delay and the setup time of a register in your formulas for the latency of this adder.

Task 2

Draw a graph in MS Excel showing the latency of this adder as a function of w for the following values of n and the range of values for w .

- $n=512, 1024, 2048$
- $w=8, 16, 32, \text{ and } 64$.

Task 3

Find analytically an optimal value of the word size, w , for which a 1024-bit High-Radix Parallel Prefix Network Adder has the smallest latency. Compare the results for the case of Kogge-Stone vs. Brent-Kung Parallel Prefix Network.

Implementation Problem

Task 1

Describe the High-Radix Parallel Prefix Network Adder (without carry in) in VHDL or Verilog.

Use generics (in VHDL) or parameters (in Verilog) for the operand size $n=2^k$, and the word size w . Use "+" for the description of ripple-carry adders.

Your code should allow the choice between two possible types of Parallel Prefix Network: Kogge-Stone and Brent-Kung.

Task 2

Surround your adder with a w -to- n SIPO (Serial-Input-Parallel-Output converter) for the inputs A and B , and an n -to- w PISO (Parallel-Input-Serial-Output converter) for the output R .

Store c_{out} in a D flip-flop.

Task 3

Synthesize and implement your circuit for $n=1024$, $w=16$, 32 , and 64 , and two Parallel Prefix Network types, targeting Virtex 6 FPGAs.

Draw diagrams showing the dependence of the latency on the value of w , separately for each of the Parallel Prefix Network types.

Hint: *Assume that the circuit latency is equal to the minimum clock period returned by the tools.*