life.augmented

STMicroelectronics

Computer Arithmetic Training

Nov.2024

LAB 3

Designing and Testing Variants of Multipliers

Instructor:
Ahmed Abdelsalam

Due to:

Sat 22 Feb.2025,11:59PM

Objective

The objective of this lab is to ensure we have a good understanding of the architecture and operation of various types of fast Multipliers, and to compare them in terms of cost and delay using Verilog/VHDL. Additionally, we will verify their functionality through a testbench.

Prerequisites

- Good understanding of digital logic design.
- Familiarity with VHDL or Verilog HDL.
- Good Understanding for booth encoding and CSA Trees.

Notes for submission

1. Upload Verilog/VHDL files to GitHub (or compress them as Zip file and upload to Drive)
2. Report (PDF format) contains all the requirements specified below

Requirements

- **RTL** the Verilog/VHDL code implements the assigned question and comment your design well upload all the code files and attach the link to the report
- **Simulation** write simple Testbench covers common cases and run simulation attach screenshots to show the output or waveform
- **FPGA** run synthesis and Implementation using VIVADO Toolchain targeting any series 7 board/device report the Utilization and critical path delay calculation
- **[Bonus] Parametrized RTL** write generic code can be used to generate N*N-bit Multiplier (where N is Parameter/Generic) of type similar to the Type in the Assigned Question
- **[Bonus] ASIC** run Synthesis and PnR for the design using any technology you have access to it. You need to specify the technology used in report. You also need to report Area, Delays and power information and attach screenshots to report.

Question1

You are asked to design a multiplier to multiply two 32-bit signed numbers using right shift algorithm (reuse the lower part of the product for multiplier) attach all the required captures and codes

Question 2

You are asked to design a radix-8 multiplier to multiply two 32-bit signed numbers using **only one** of the following algorithms (consider stored carry form)

a) Booth recoding
b) CSA Partial Tree

attach all the required captures and codes

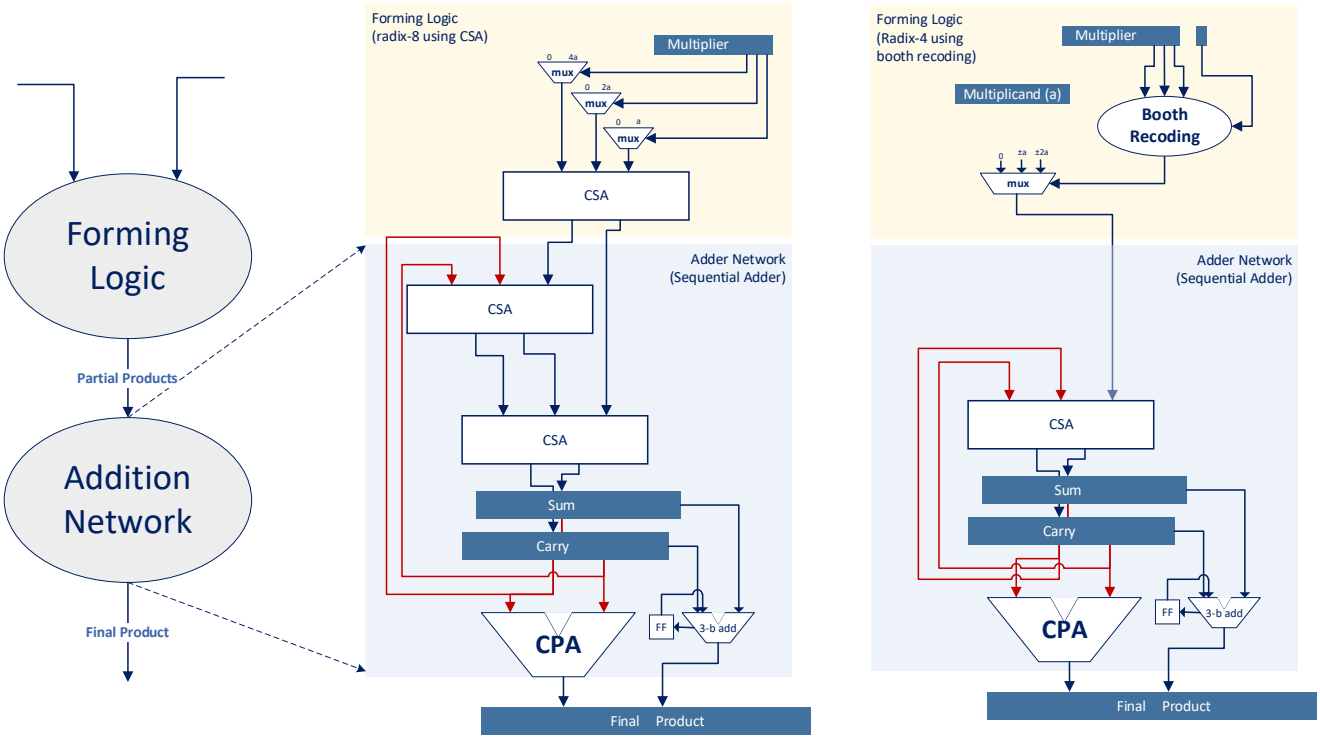

Figure 1

Fig.1 shows 2 different sequential Multipliers both are using adder network topology but the way they form the partial products are different left one uses booth recoding, and the right one is using 3-2 compressor to form it.

Hint: CPA can be simple “+” operator it’s better to uses it inside a module so we can analyze the utilization report

Question 3

You are asked to design a parallel multiplier to multiply two 32-bit signed numbers using **only one** of the following option for the Addition Network

a) Wallace/Dadda Tree
b) Parallel counter 64-2
c) 4 to 2 compressor Tree
d) Array Multiplier

attach all the required captures and codes

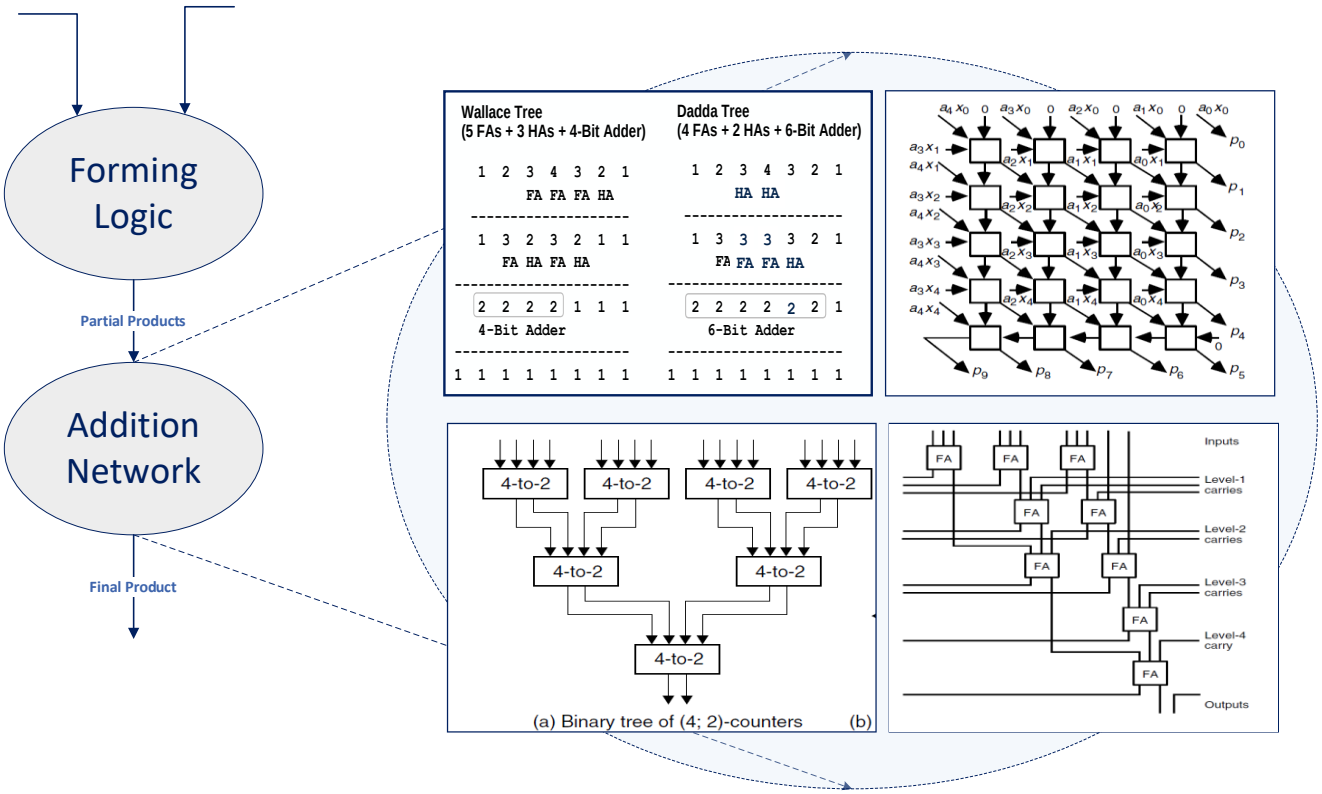

Figure 2

Fig 3: shows the general equations that can be useful to build a generic Wallace/Dadda Tree left one is Wallace, and the right one is Dadda

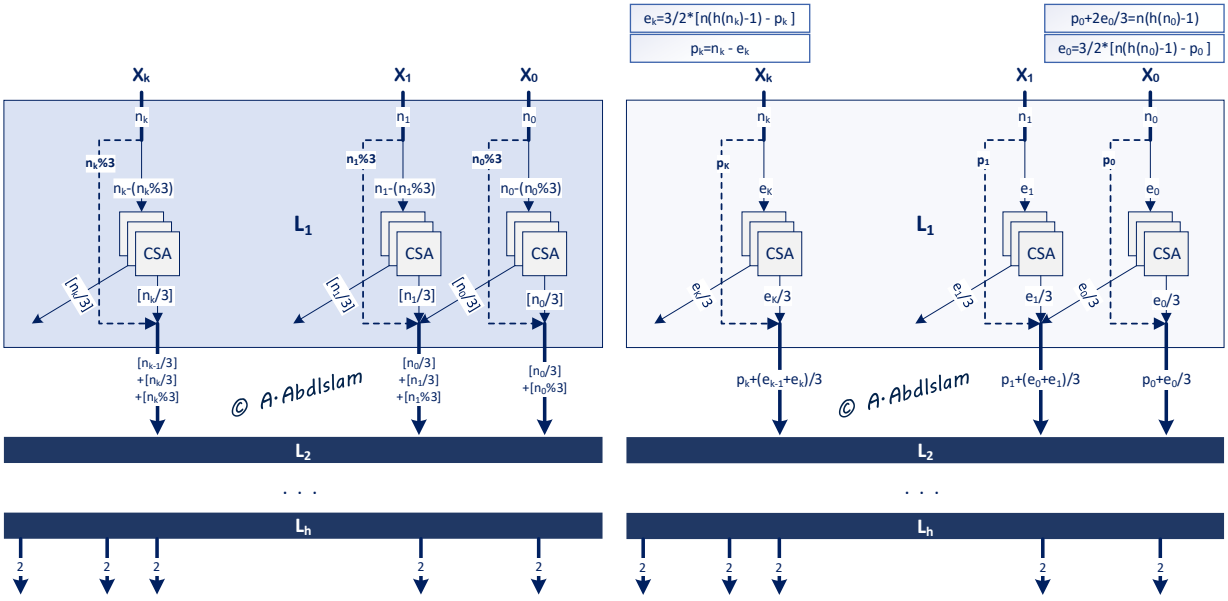

Figure 3