The Matrix-Matrix Systolic Array: Design, Simulation, and Implementation

Group # – AEP227 Github Name, Teammate 1, Teammate 2

Abstract – The Matrix-Matrix Systolic Array uses hierarchical, modular design to break down the design process of the complex matrix multiplier into a series of smaller designs. This project required us to recognize general patterns in symbolic, testbench, and layout-level design to streamline this project into a short timeframe.

Introduction

In Lab 2, we realized several general principles in the Compound Gate. This is, there is no one way to approach a complex design and they can greatly vary in design. These principles, just because the Systolic Array is more complex in its logic, never stop applying. Our team approached the Systolic Array by making our designs modular and accessible and sharing the load to implement our design. Our process is as recommended throughout our education – Design, Simulation, and Implementation.

The Systolic Array's design is a bottom-up hierarchical design. This is, our team first designed the subsystems as to allow the Data Processing Unit (DPU) and Systolic Array to function. Figure 1 has the graphical representation. Our discussion begins with the modules at the bottom.

Figure 1 - Organizational chart of the Systolic Array

As an assumption, we assume familiarity with the Unit Inverter, Double Inverter, D Flip-Flop (DFF), and the 4-bit Ripple-Carry Adder (RCA) as designed in Labs 1-4. In all these cases, the prescribed W/L = 5/3 ratio is used throughout. The new designs are described below.

Design and Technical Approach

First, consider the registers, 12-bit RCA, AND2, and the 4x4 Multiplier. Each of these designs was created by concatenation. This is, an existing design – the DFF, the 4-bit RCA, NAND2 (Lab 1), and the Unit Inverter – were combined to create these devices. While this strategy of concatenation continues throughout our design, these are fully combinational or memory designs. This allows for ease of testing, which will be further discussed in the Simulation section.

Of these devices, the 4x4 Multiplier is perhaps the most interesting. It was made completely out of existing structures (3x4-bit adders and AND gates). This was the first modular design that took advantage of the input/output pin placement of previous modules. With inputs on the top of the adders and the outputs along the bottom, it was simple to stack the individual pieces as

tightly as possible and connect the I/O horizontally.

These devices created allow for the DPU to function. As specified, the DPU accepts two four-bit inputs A,B. These inputs are sent to their respective 4-bit register which emits Aout, Bout on the next positive clock edge. A,B also provide input to combinational

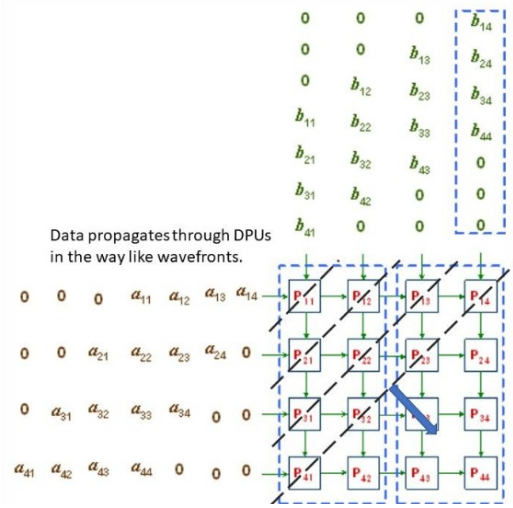


Figure 2 - Illustrated input sequence, as demonstrated from the assignment page.

arithmetic logic. Upon entry, A,B provide stimulus to the 4x4 Multiplier. The 8-bit product output is then provided to a 12-bit adder (the 8-bit product is zero-filled to 12 bits in our adder's design). The adder's output then enters an array of buffers, as the adder's drive strength is insufficient to overpower the present state of the DFFs. This register's value represents the product matrix's cell's value. Each clock cycle computes a new partial product from the stimulus A,B values.

The Systolic Array is a fully sequential system that receives a single 4-bit A,B input value each clock cycle using the prescribed input sequence per bit, as shown in Figure 2. This arrangement propagates the matrix in sync with matrix multiplication. On every positive clock edge, another value of the matrix or a zero-filled value will enter. The Matrix-Matrix Multiplier allows for a 4x4 A,B as input.

This is the design of the Systolic Array.

Implementation: Layout Design

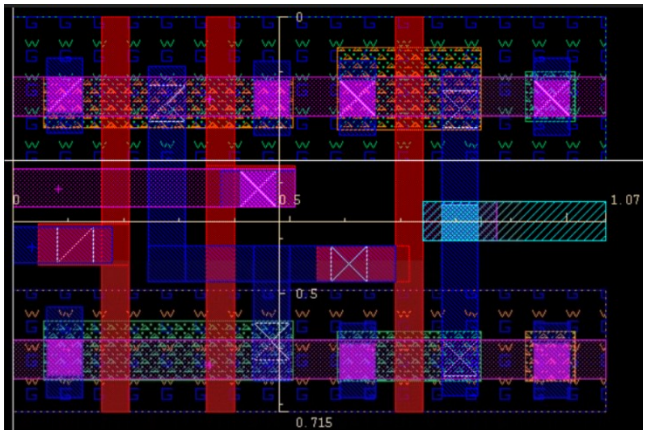


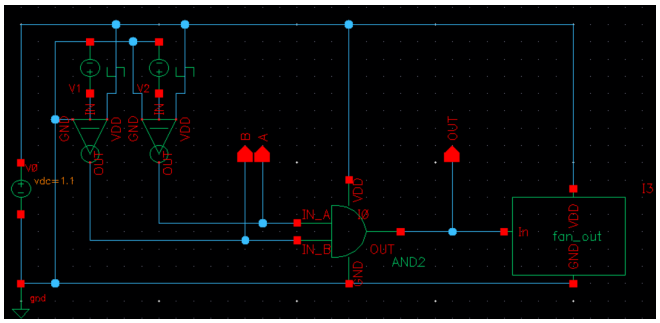
Figure 3 - AND2 Layout. The left-most Metal2, center, has the A pin. The left-most Metal1 has the B pin. The right-most Metal3 has the OUT pin.

Every layout design was intended to be used in a larger system. A prime example of this was the 12-bit ripple carry adder. All the inputs were Metal3 inputs along the top of the module and the outputs were all Metal3 along the bottom. While this leads to very wide layouts, it provides for fast and cheap (manhours) implementation. No hunting or getting confused of which input/output is which.

This approach gave the biggest time-saving in the layout of the array. The DPU layout had pins in locations such that it was a matter of simply resizing the Metal3 A inputs and Metal2 B inputs to meet up with the DPU below/to the right. With the VDD, GND, CLK, and CLK_N inputs along the top of each DPU, rails carrying each of those

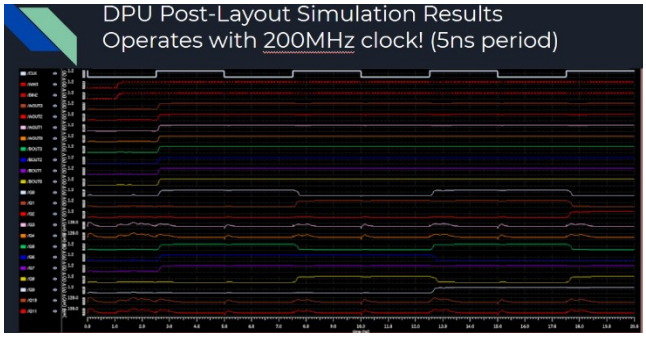
signals were run horizontally along the top of each row of DPUs. As a testament to this design philosophy, the entire DPU array layout took about an hour and a half to construct.

Every design is tested using a standard



testbench. This is, there is an array of vbits and vpulse (for clocks). These provide the input stimulus to the device under test, centered. The output signals then all have a capacitive load with FO4 of unit inverters. This is summarized in Figure 4.

The Addendum, attached, has pre-layout simulation of each subsystem module (all but the Systolic Array and DPU). Discussed in this section, most importantly, is the DPU post-layout simulation and the Systolic Array pre-layout simulation.



The DPU post-layout trace is in Figure 5. This figure demonstrates that, for a 8ns clock with a pulse width of 4ns, the DPU functions with up to a 200MHz clock. The

More details are in the Addendum, but these are the major area statistics. The DPU has area 901.1572um^2. The Systolic Array has area 14.1761mm^2.

Simulation: Testing Approach

combinational propagation delay (Addendum: multiplier) and setup time of the 12-bit register (Addendum: register) has that about **1ns is required for the inputs to propagate. This has a switching frequency of about 142.8571*10^6 bits/sec.**

We attempted to run the Systolic Array's pre-layout simulation, but experienced the same problem as Group 4 in that the testbench stalls and takes too long to calculate the matrix multiplication. It is our belief, however, that the DPU's timing is independent of its integration into the array and still holds a **8ns period (200MHz) and 142.8571*10^6 bits/sec throughput.**

Conclusion and Summary of Challenges and Innovations

We have established the following truths and strategies. In design, modularity and hierarchical design provide for ease in implementing and integrating a schematic and layout design into larger systems. In the layouts, all pins intended for external connections are made very accessible to the design at large. This sacrifices area for ease of integration.

Hindering our development was two critical issues. The Systolic Array passes DRC, but fails LVS due to an unexpected short and pin translations. Previously, this array failed DRC because of antenna errors (too much metal-to-poly ratio in certain areas of the layout). Additionally, our testbench takes

too long to compile to iterate our testing design.

We have overcome, however, drive strength issues within the DPU and establishing a workflow of “sharing the load” between our three members. This allowed for us to complete this project to this level of sufficiency.

Addendum

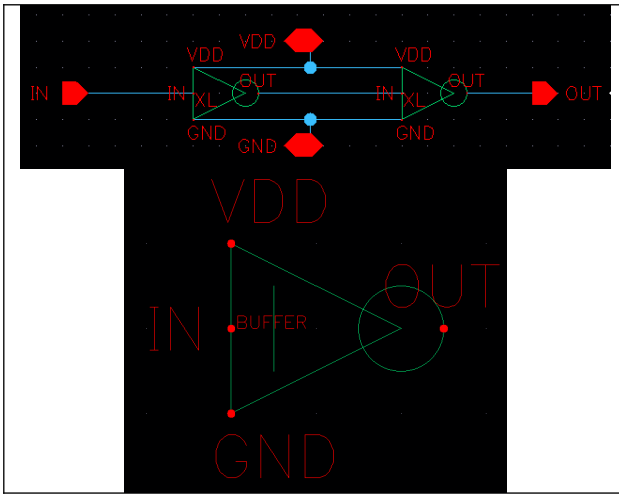
Design

1. Building Blocks
    a. Buffer
    b. AND2
    c. Fanout
2. Data Processing Unit (DPU)
    a. 4-bit and 12-bit Registers
    b. 12-bit Ripple-Carry Adder (RCA)
    c. 4x4 Multiplier
3. Systolic Array – Matrix-Matrix Multiplier

Design: Building Blocks

These components simplify common devices for use in input distribution and testbench design.

Buffer



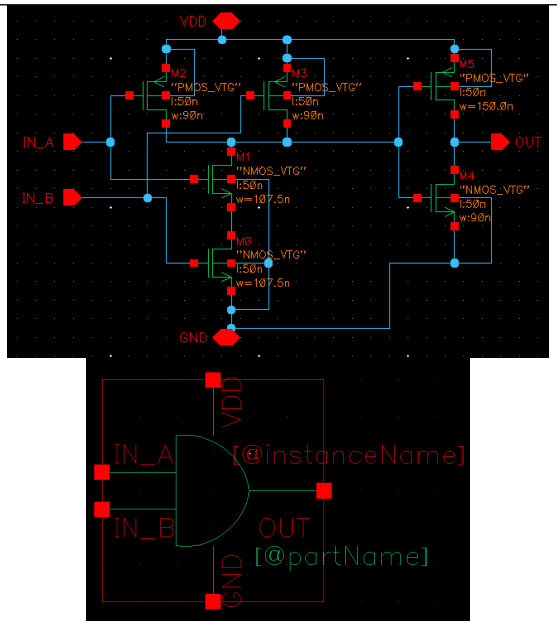
Pinout

- Inputs
  - IN
- Outputs
  - OUT
- Jumpers
  - VDD
  - GND

Components

- 2x Double Inverters
  - NMOS_W = 300n
  - PMOS_W = 180n

AND2



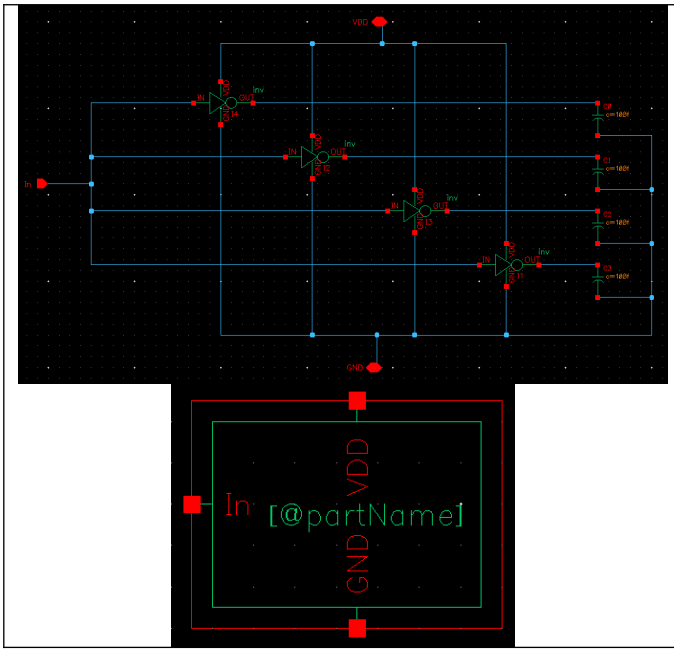
Pinout

- Inputs
  - IN_A
  - IN_B
- Outputs
  - OUT
- Jumpers
  - VDD
  - GND

Components

- NAND Gate
  - PMOS_W = 90n
  - NMOS_W = 107.5n
- Unit Inverter
  - PMOS_W = 150n
  - NMOS_W = 90n

Fanout



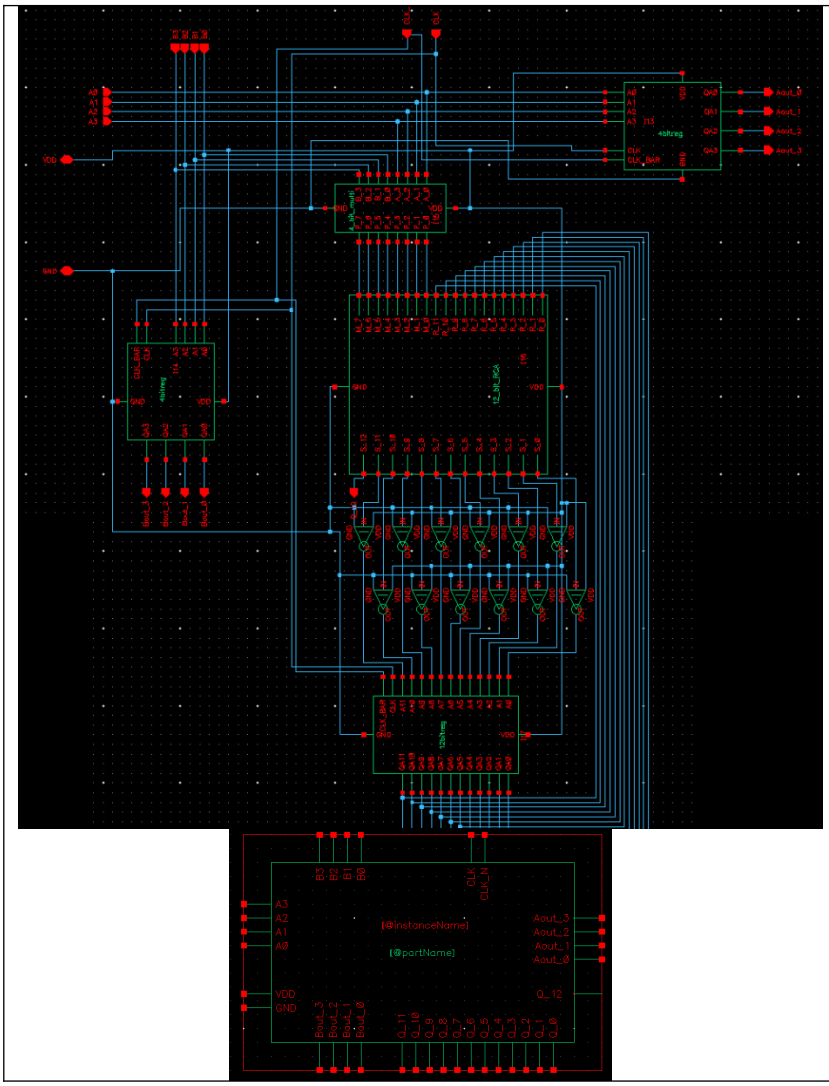
Pinout

- Inputs
  - IN
- Jumpers
  - VDD
  - GND

Components

- 4x Unit Inverter
  - PMOS_W = 150n
  - NMOS_W = 90n
- 4x 100fF Capacitors

Design: DPU



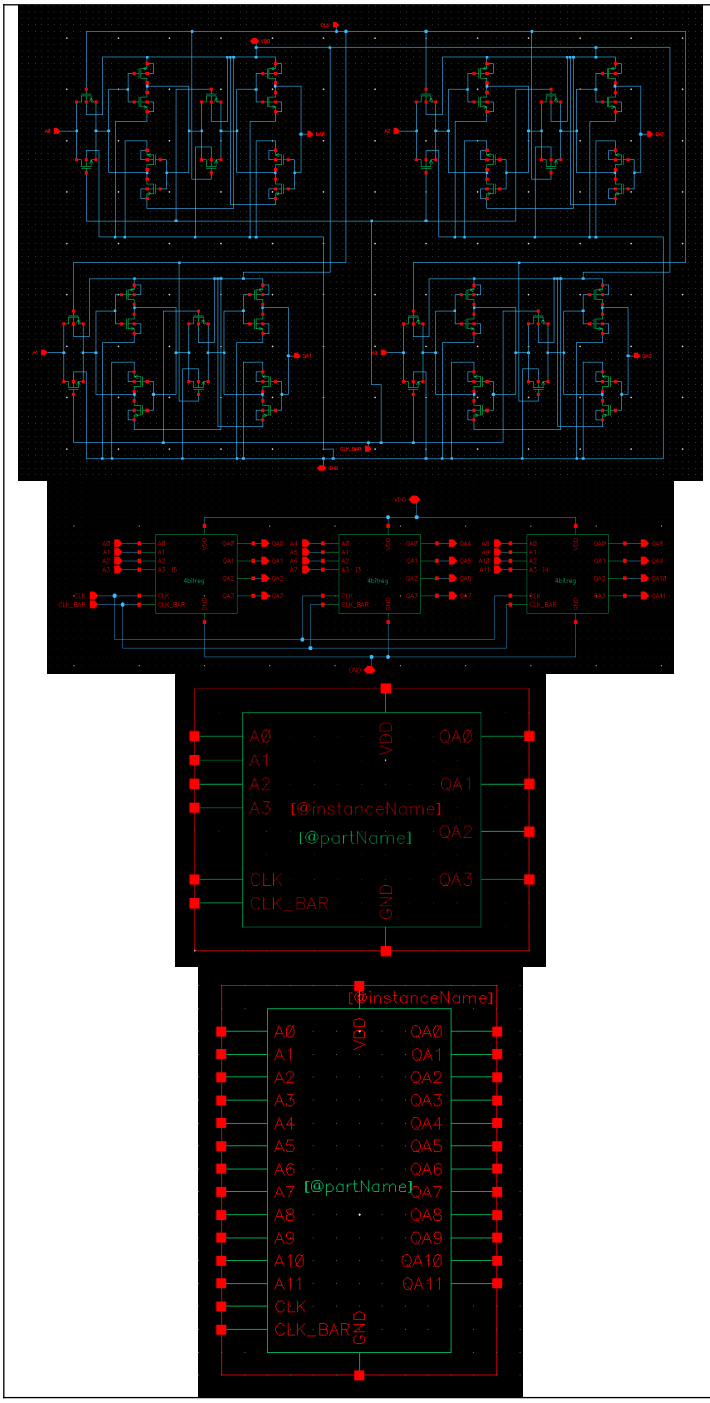
Pinout

- Inputs
  - A[3:0],B[3:0] – Combinational input
  - CLK,CLK_N
- Outputs
  - Q[12:0] – Sum of Products register
  - Aout[3:0], Bout[3:0] – 4-bit register passes these to next DPU on clock edge
- Jumpers
  - VDD
  - GND

Components

- 2x 4-bit Register
- 1x 12-bit Register
- 1x 4x4 Multiplier
- 1x 12-bit Adder
- 12x Buffer

Registers



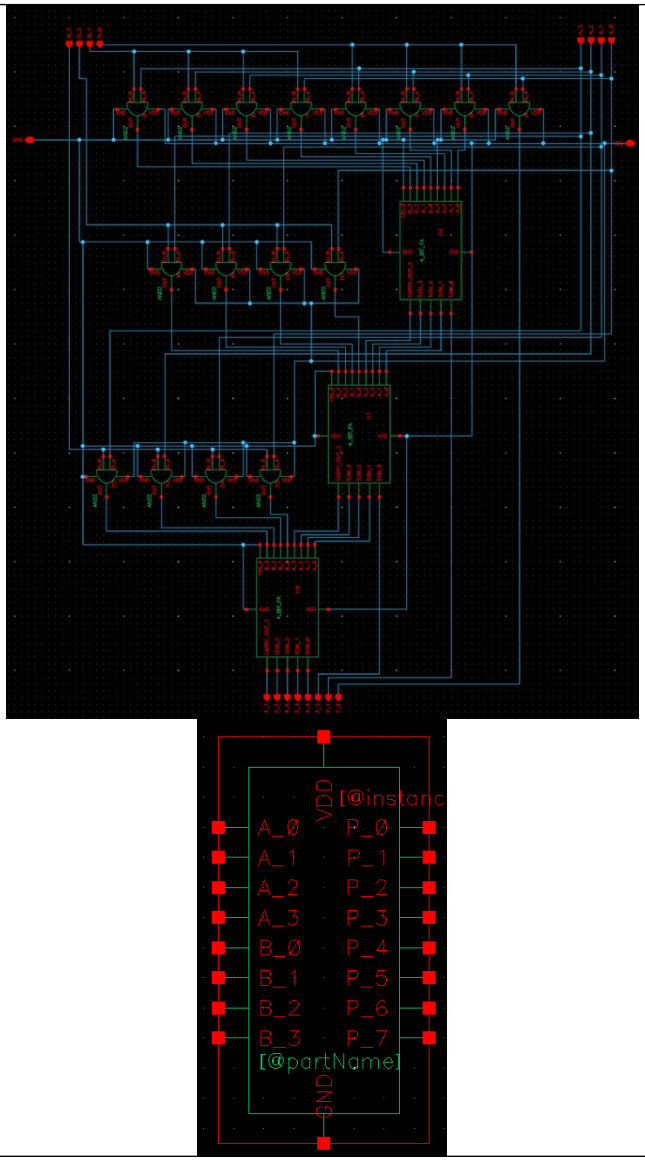
Pinout

- Inputs
  - 4-bit: A0-A3
  - 12-bit A0-A11
- Outputs
  - 4-bit: Q0-Q3
  - 12-bit: Q0-Q11
- Jumpers
  - VDD
  - GND

Components

- 4-bit: 4x DFF
- 12-bit: 12x DFF

Multiplier



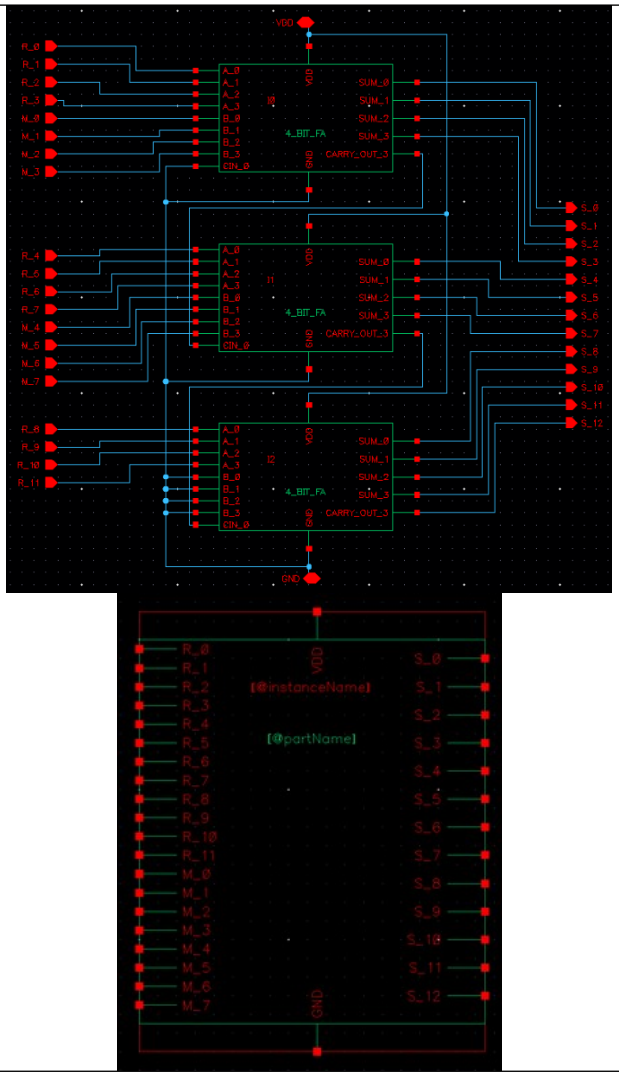
Pinout

- Inputs
  - A_0 – A_3
  - B_0 – B_3
- Outputs
  - P_0 – P_7
- Jumpers
  - VDD
  - GND

Components

- 3x RCA
- 16x AND Gates
  -

Adder



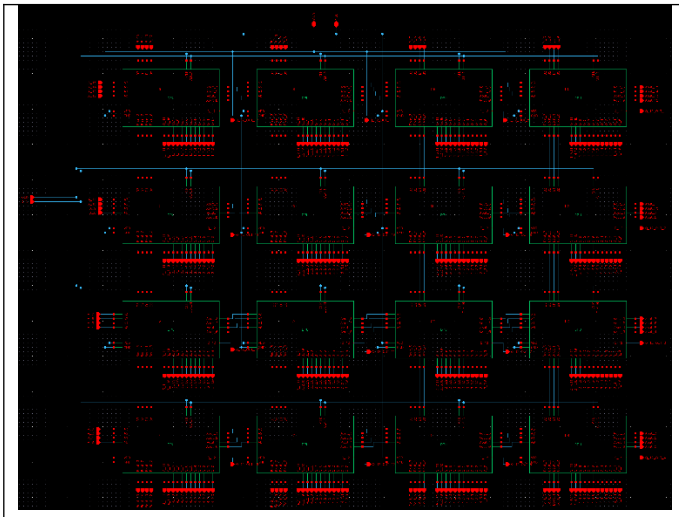
Pinout

- Inputs
  - R_0 – R_11 (from register)
  - M_0 – M_7 (from multiplier)
- Outputs
  - S_0 – S_11 (sum)
  - S_12 (Carry out, unused)
- Jumpers
  - VDD
  - GND

Components

- 3x 4-bit RCA

Design: Systolic Array (Matrix-Matrix)



Pinout

- Inputs
  - A_0 – A_15
  - B_0 – B_15
  - CLK,CLK_N
- Outputs
  - i_j.0 – i_j.11 where is the row number and j is the column number
- Jumpers
  - VDD
  - GND

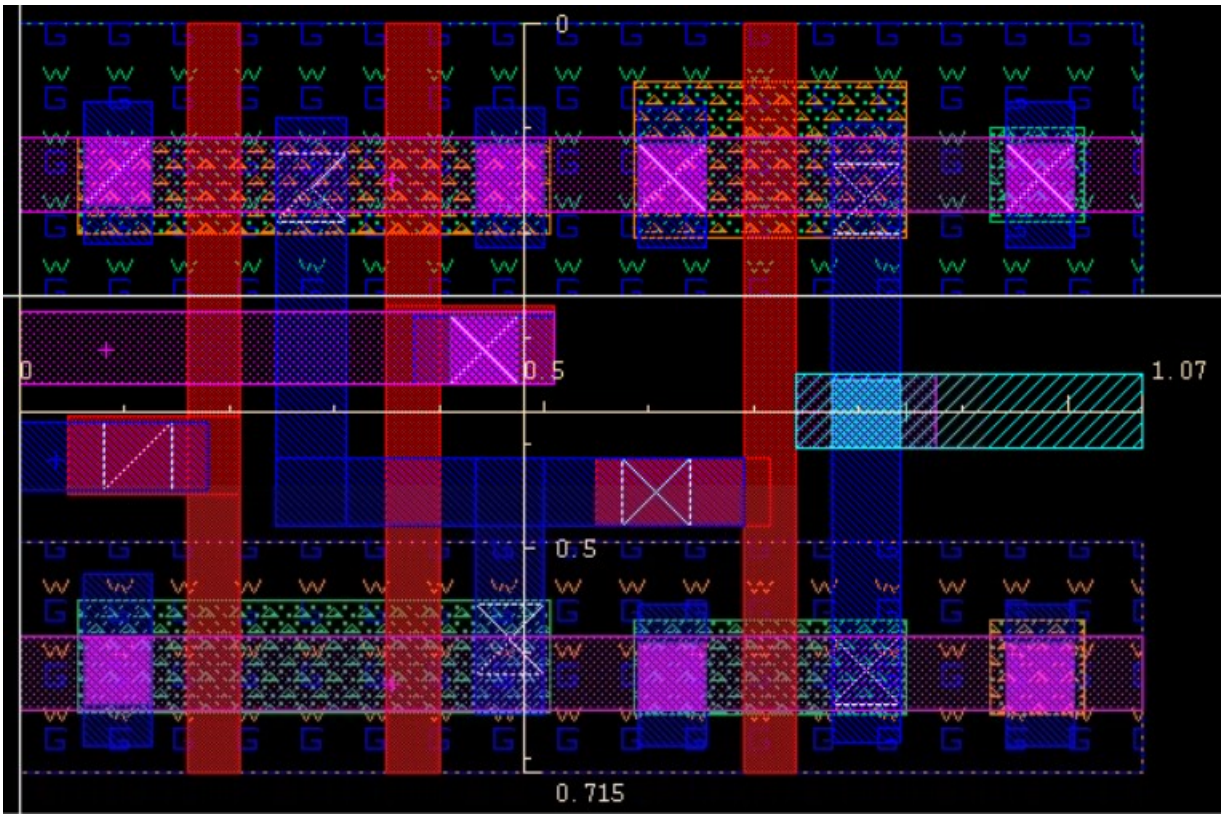
Components

- 16x DPU (registers enclosed)

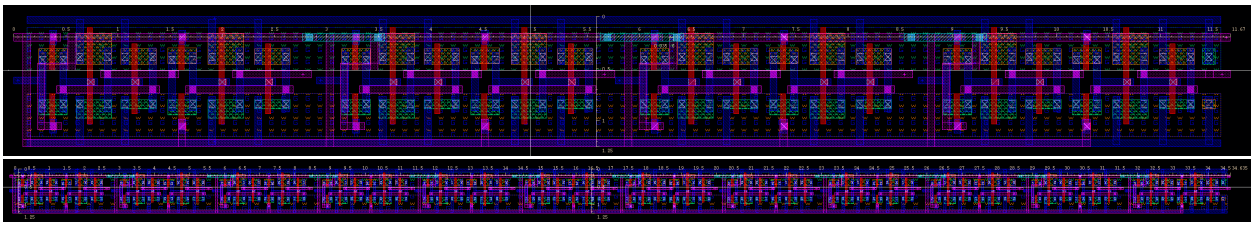
Implementation

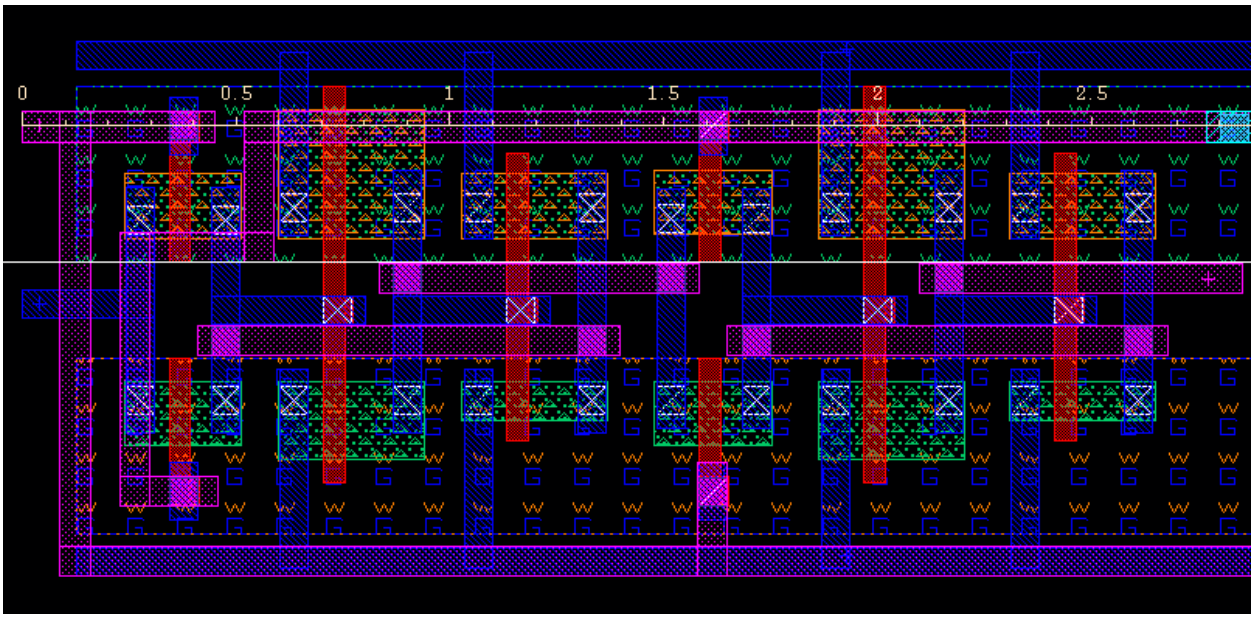
DRC, LVS, and PEX are only shown for the DPU and Systolic Array (DRC and LVS only). Excepting the Systolic Array, all subsystem layouts pass these and the DPU's passage implies this.

AND2



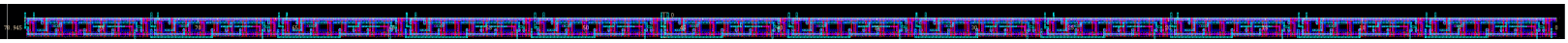
Register

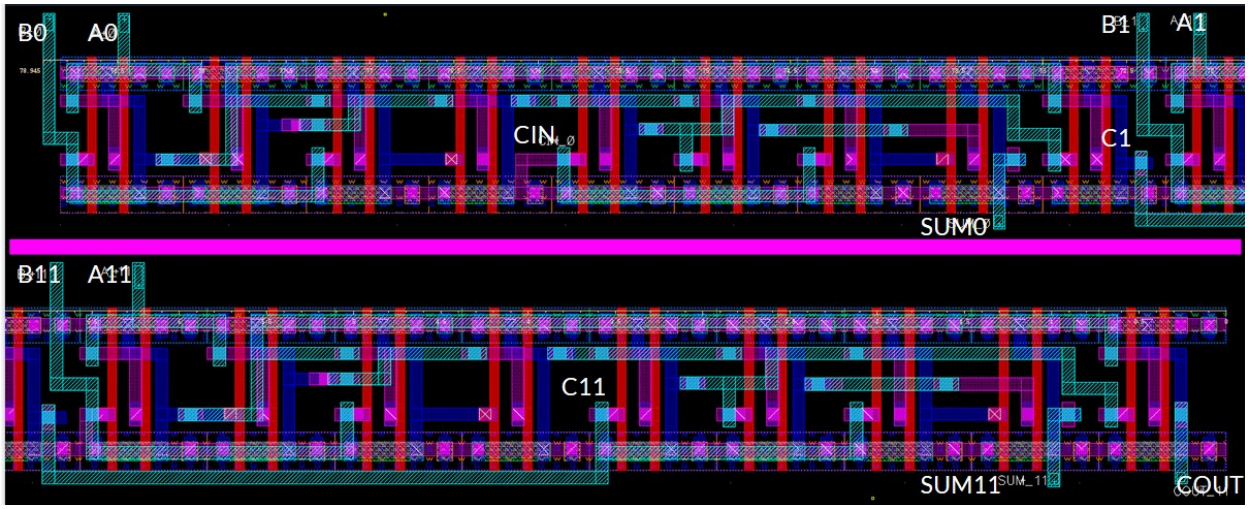




One DFF cell is shown, above, separate from the layout. Each DFF is independent, excepting for their VDD and GND inputs.

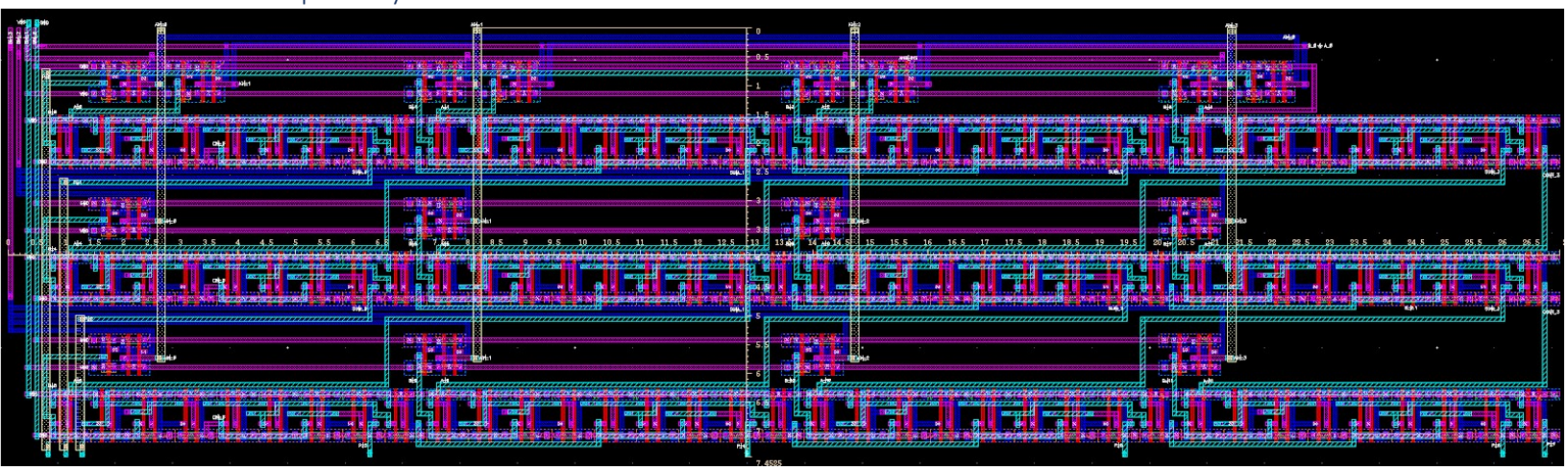
12-bit RCA



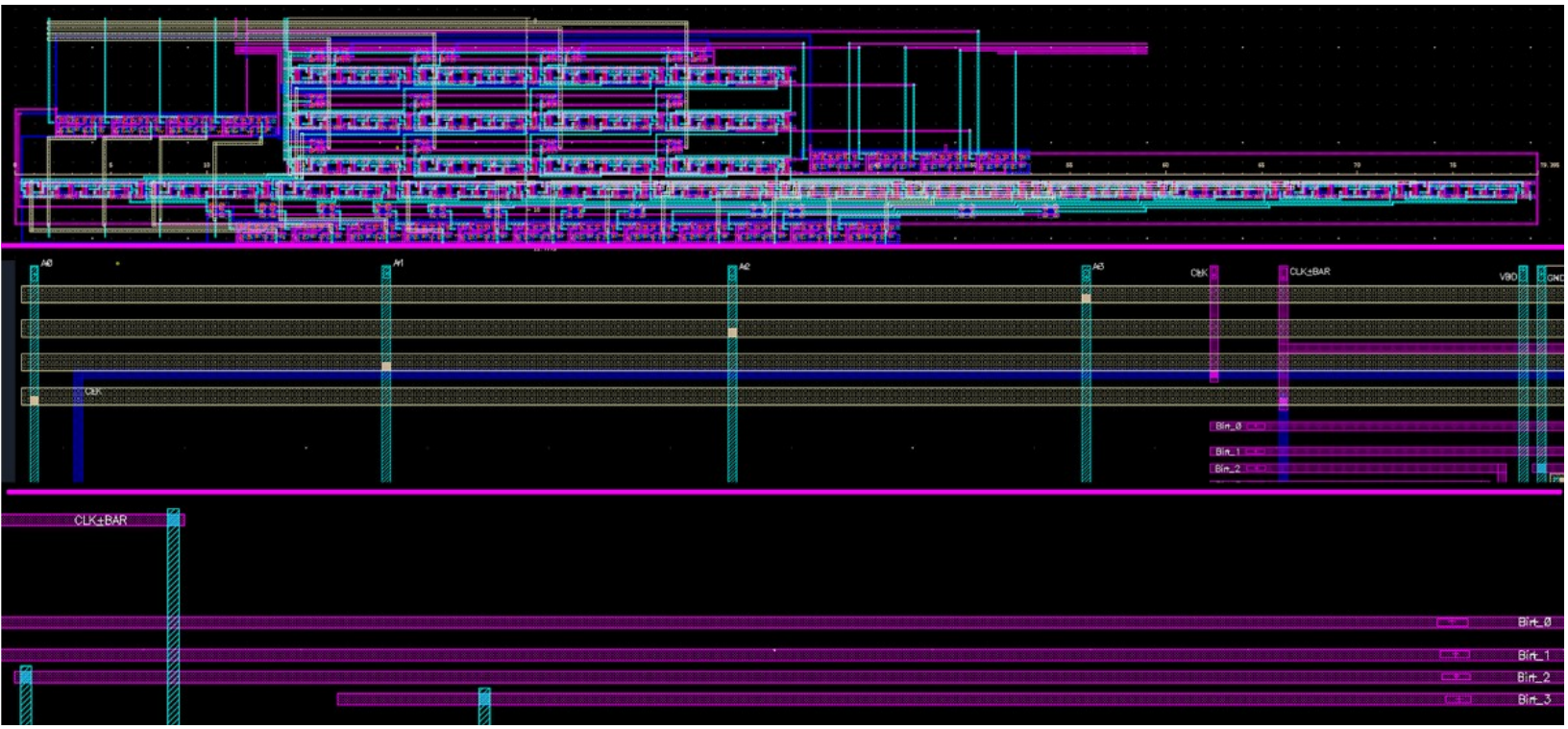


Shown above are two RCA cells.

4x4 Multiplier Layout

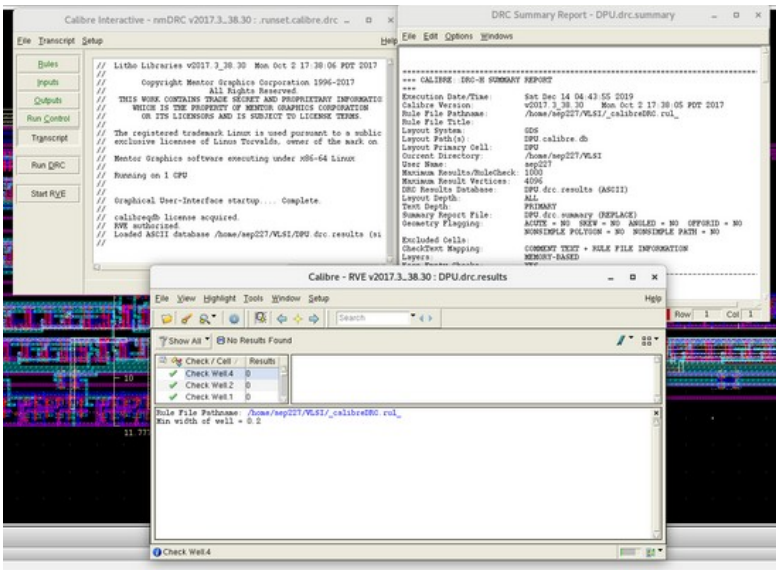


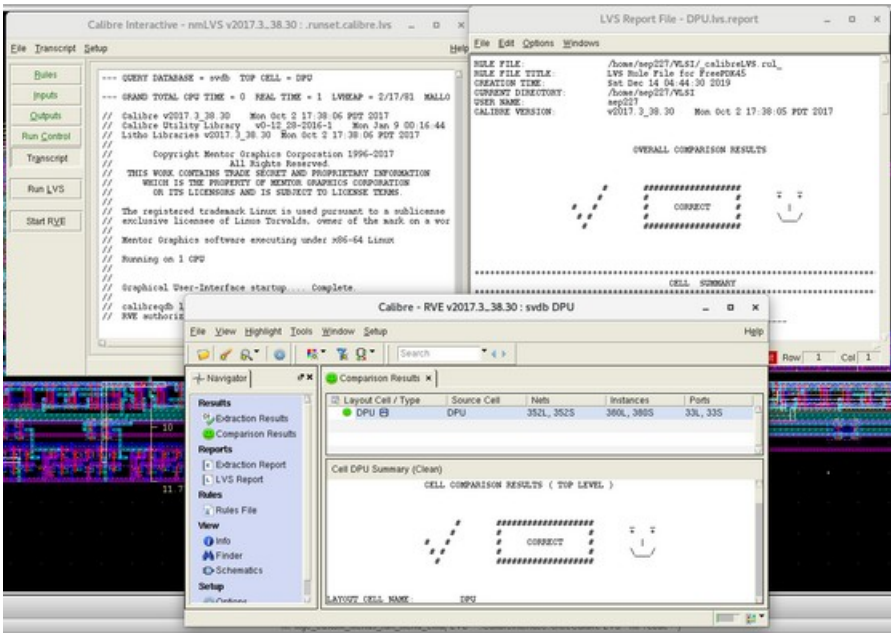
DPU

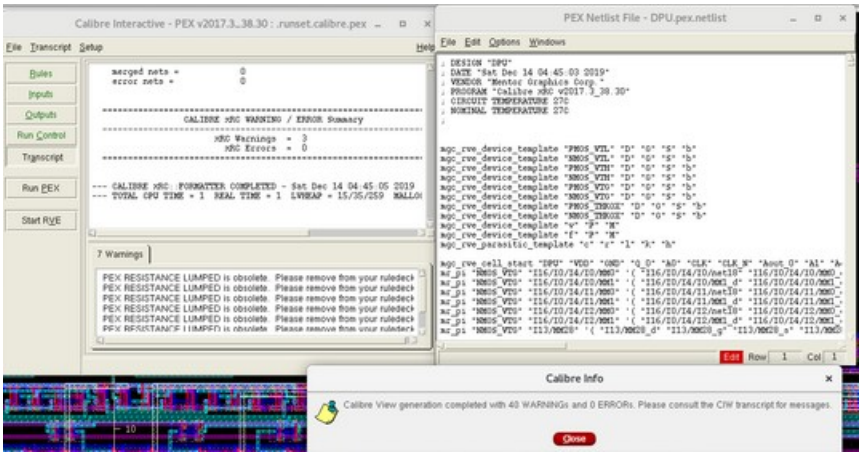


The first third in the above figure has the DPU layout. Stacked are the 4-bit registers and the multiplier, the 12-bit RCA is the long piece in the middle, and the lower pieces have the 12-bit register and buffers. Shown in the lower two thirds are the input pin locations.

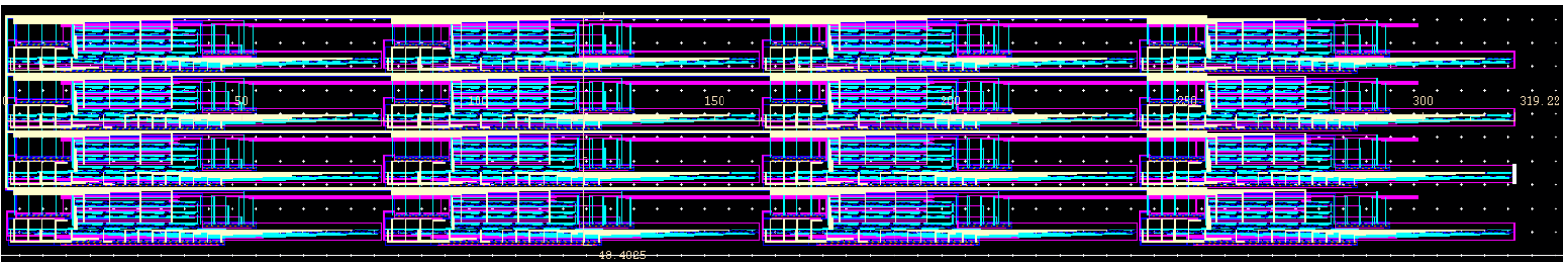
Area = 901.1572um^2


Calibre Interactive - nmDRC v2017.3_38.30 : .runset.calibre.drc
DRC Summary Report - DPU.drc.summary
Calibre - RVE v2017.3_38.30 : DPU.drc.results
Run DRC
Start RVE

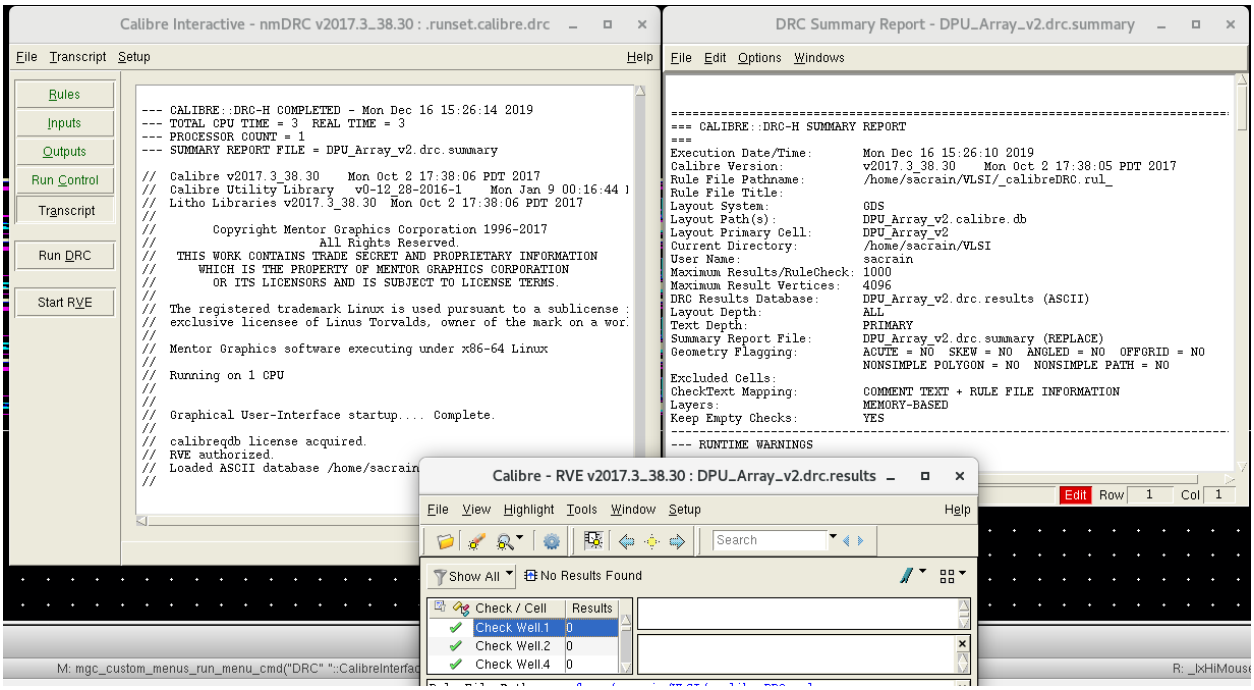

Calibre Interactive - nmLVS v2017.3_38.30 : .runset.calibre.lvs
LVS Report File - DPU.lvs.report
Calibre - RVE v2017.3_38.30 : svdb DPU
Run LVS
Start RVE

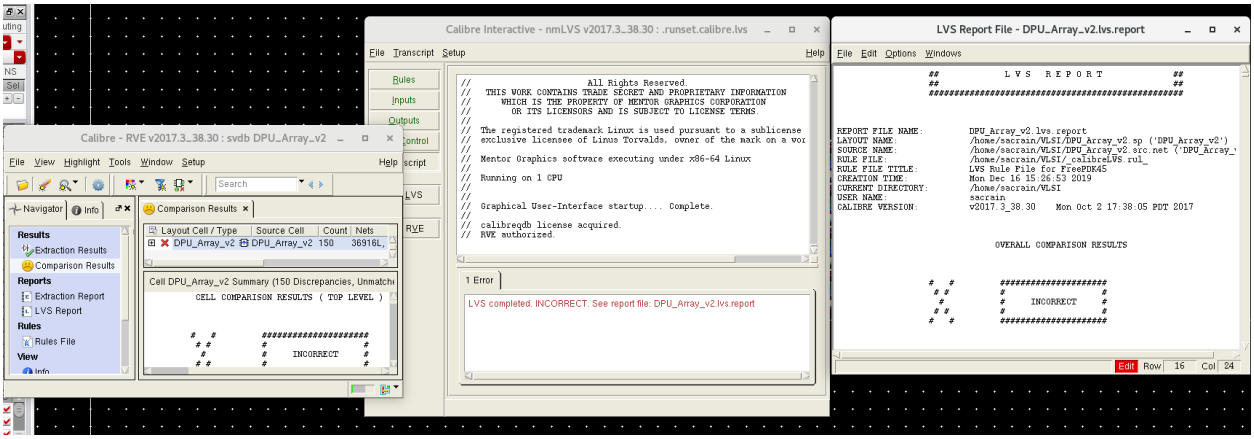

Calibre Interactive - PEX v2017.3_38.30 : .runset.calibre.pex
PEX Netlist File - DPU.pex.netlist
Calibre Info
Run PEX
Start RVE

Systolic Array



Area = 14.1761mm^2

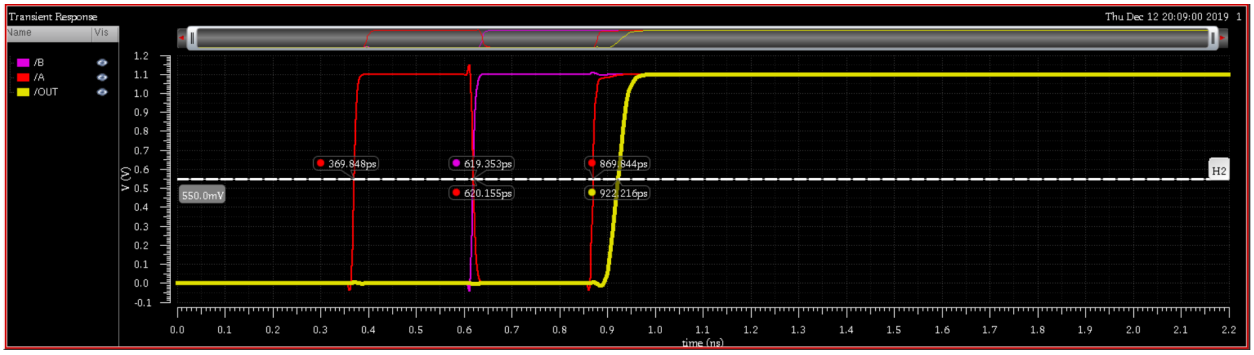




Simulation

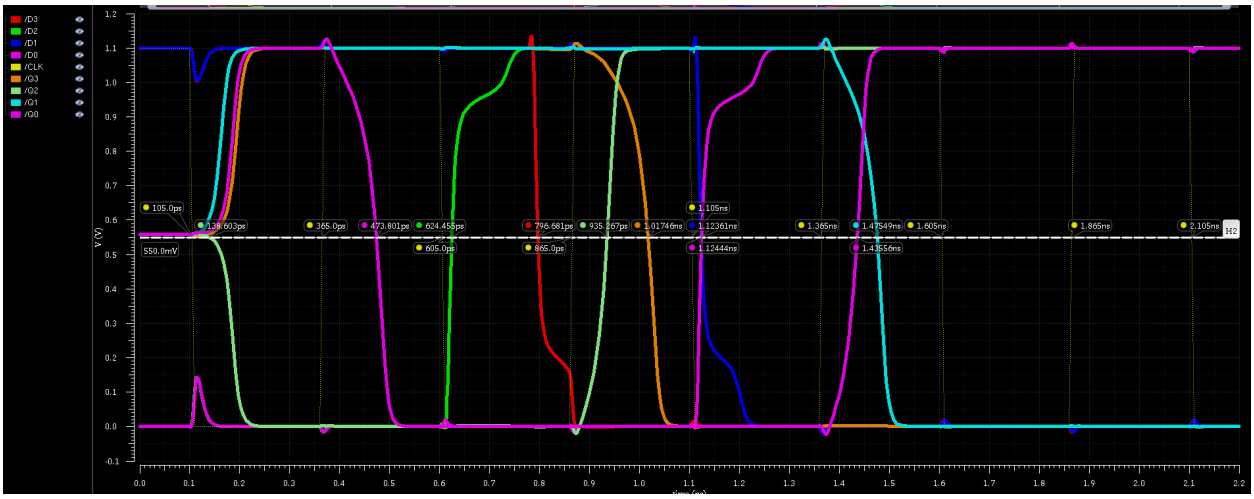
Buffers, Fanout, DFF, 4-bit RCA are all confirmed working from testbench operation or from previous labs. **All included simulation traces are post-layout.**

AND2



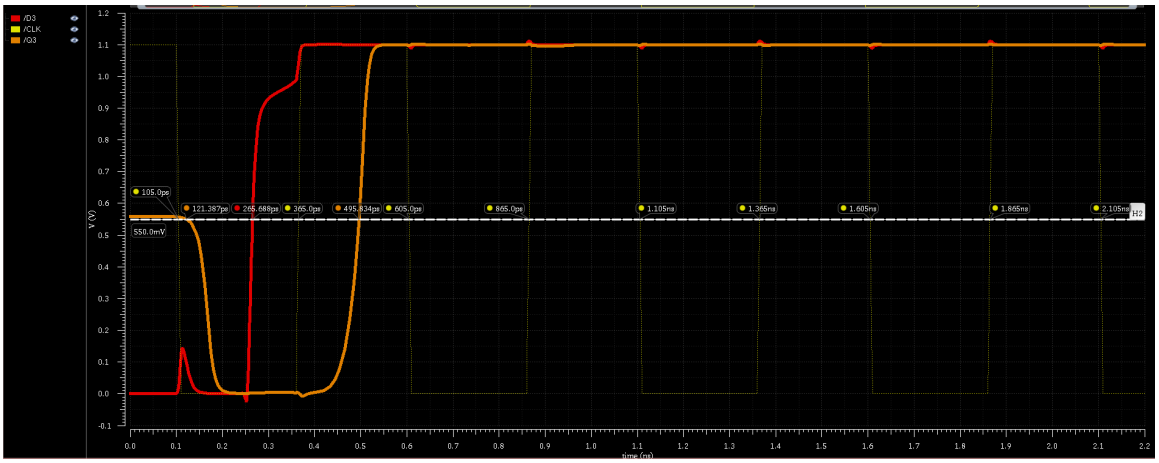
AND2 is only tested for functionality.

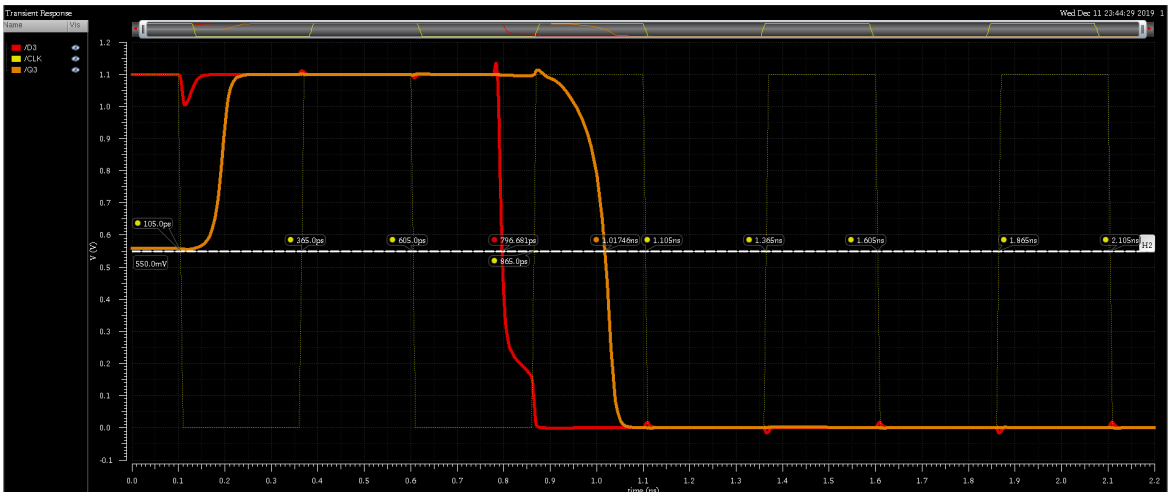
4-bit Register



Observe

- Propagation High-to-Low: 110.49ps
- Propagation Low-to-High: 70.56ps

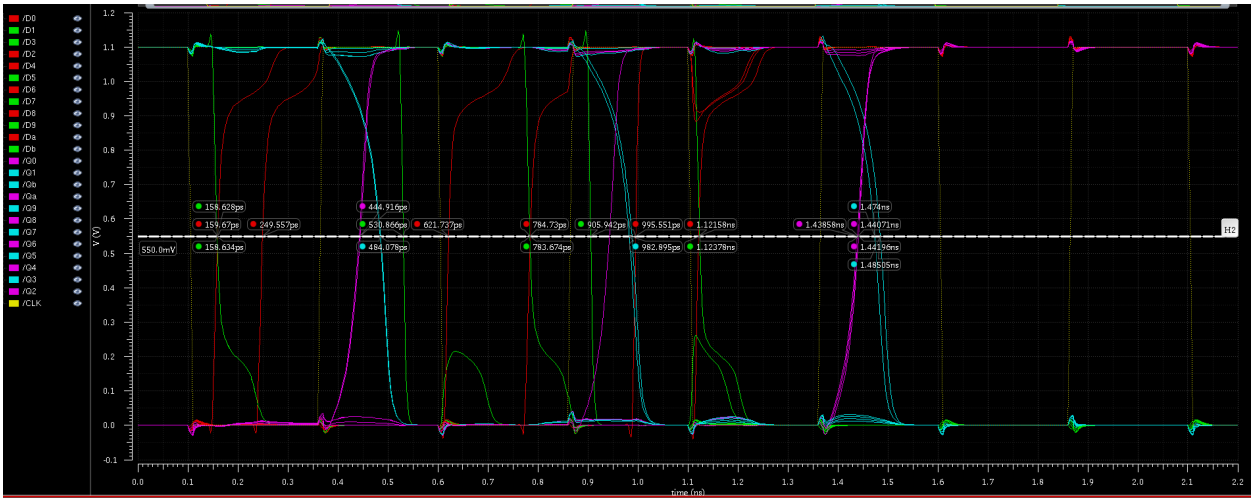




Observe

- Setup_HL: 68.319ps
- Setup_LH: 99.312ps

12-bit Register

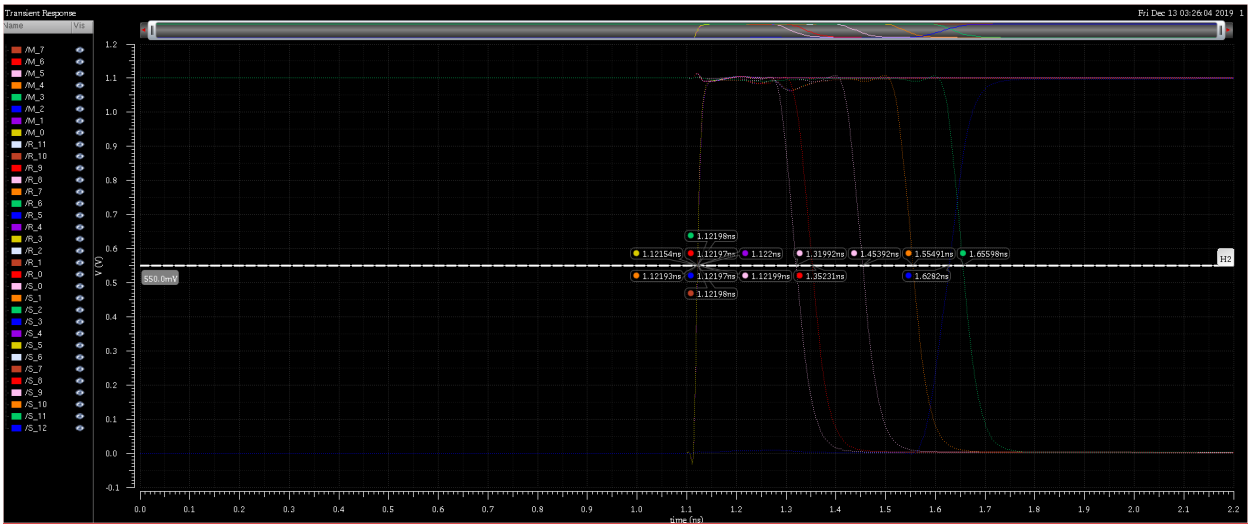


Observe, for example

- Propagation Delay Low-to-High: 78.685ps
- Propagation Delay High-to-Low: 116.875ps

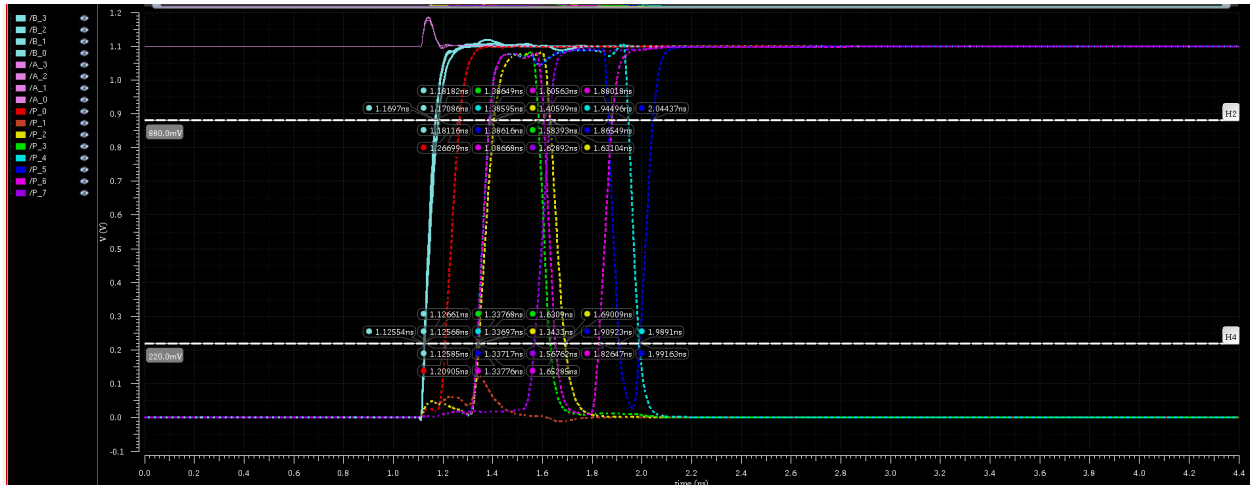
Compared to our system and clock speed at large, this is nearly a negligible increase in delay (~10ps). The observed setup and hold times are exactly those of the 4-bit Register.

12-bit Adder



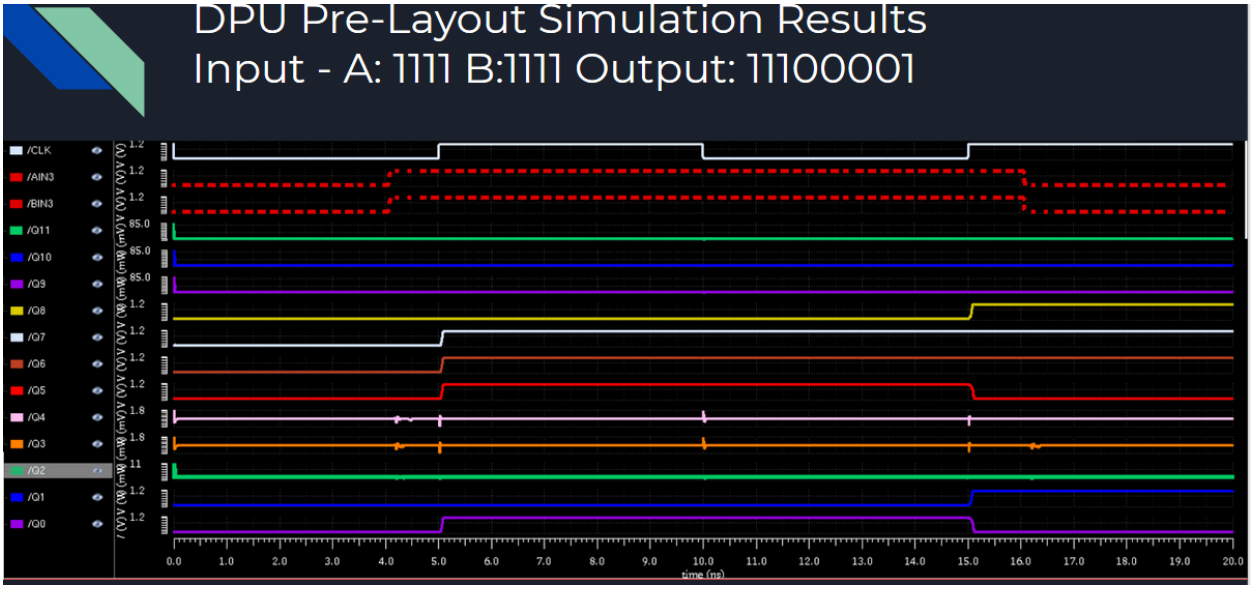
The adder is tested for functionality and propagation delay of the final bit. This is **534.44ps.**

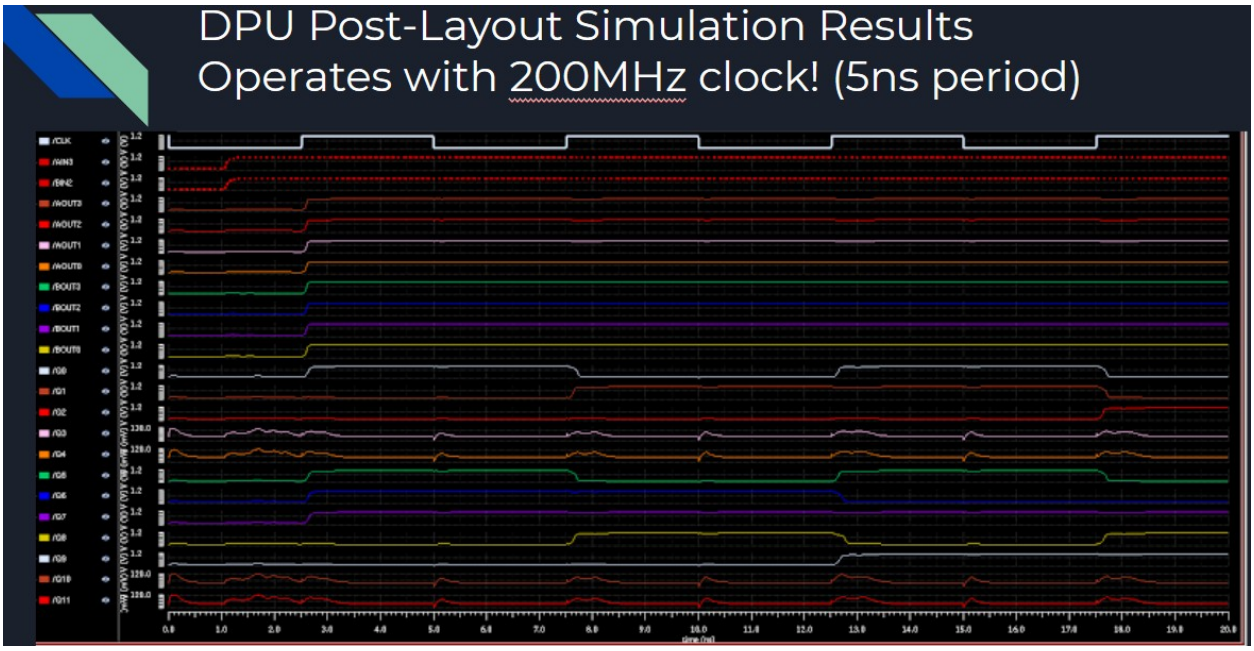
4x4 Multiplier



The multiplier is tested for functionality and propagation delay of the final bit. This is **918.83ps.**

DPU





As demonstrated, the DPU functions correctly up to a clock rate of 200 MHz.