ENGG4550 Project Report

Group 29

Members:

Charlie Magri Ricardo Quadras

Student IDs:

1194399 1092724

April 6th, 2024

Introduction

In this project an 8-bit program counter will be designed from the ground up from scratch using components created and provided through cadence modelling software. All higher-level blocks will be created from the following components:

- Nmos Transistor
- Pmos Transistor
- Voltage Source
- Ground
- Capacitor (technically not-needed for top level but included)
- Wire (default from Virtuoso, both thick and thin)

All other gates will be a combination of those base components to create the top-level program counter. This project is a test of the skills learned through tutorials/Labs 1-3 and provides a benchmark for our progress in VLSI design.

Objectives

To create a successful 8 bit program, counter all sub-components must be working and with enough minimized delay to efficiently compute results. A high delay will result in incorrect values or undefined behavior during transition times. To ensure this benchmark is met, the design be capable of 1Ghz of frequency

The program counter must have 2 modes of operation, Default program counting (increase by 1) and branch conditions, which will be a fixed jump. This jump will be determined by a separate adder that has a constant input.

The Program counter must also have a reset, that when triggered will reset the registers to 0. This is essential to showing restart functionality and simulating rebooting.

Finally, the purpose of designing the program counter is to compare different implementations of different blocks regarding speed, area, and power consumption. This will allow for a demonstration of the positives and negatives of each design.

Analysis, Design & Justification

Multiplexer Design

For the design of our multiplexer, we chose the simple 2 input static CMOS pass-transistor as seen in Figure 2. This was the circuit family that the most easy and simplest to implement for the multiplexer, and there are several distinct advantages and disadvantages to this approach. Firstly, a distinct pull-down and pull-up switch network for

strong 0s and strong 1s built of pMOS transistors and nMOS transistors respectively ensures that the output is always strongly driven, reducing noise and ensuring logic levels are never degraded. This also heavily reduces our static power consumption, but there is higher capacitance with the complementary transistor pairs leading to increased delay and effort. As the delay of a circuit is directly proportional to the capacitance, there is a tradeoff between switching speed and its restoring nature. Figure 3 shows the combined 8 bit multiplexer, made up of 8 two to one multiplexers.

Full Adder Design

The Adder designed was a CPL full adder. This design started as a single bit adder and cascaded to be an 8 bit wide adder. The design was completed on transistors alone, however an inverter symbol was used from previous lab to simplify buffering but it too only consisted of transistors. The total transistor count was 24 without buffers and 36 with buffers on both the carry out and out channels. The 8 bit adder was created using a ripple carry adder. This adder is small in size and fast enough for an 8 bit adder. If creating more bits or complicating the operation a more complex adder would be better. However the with a buffer implementation the design worked in all tests. Another design to consider would be the carry look-ahead adder in 11.21 of the textbook. This adder would increase the speed with a tradeoff in complexity and size. The textbook shows that this system would require an additional and, or and mux implementation for each bit of adder which gives substantial overhead for this design. Although it improves speed the current design met all constraints and is recommended for the reduced power consumption. Figure 7 shows the 8 bit ripple carry, and figure 9 shows the single bit CPL integration.

Register Design

For the register design, we chose the Dynamic inverting flip-flop built from a pair of back-to-back dynamic latches. We added on another inverter near the end of the circuit to produce a static non-inverting flipflop without the Q bar output. We chose this design due to it being robust, simple, compact and energy-efficient. This design's transmission gates rely on clock transitions to pass / block signals, thus buffering the clock edge rate would boost performance as a sharper transition would reduce the window the circuit is susceptible to noise & help with overall timing between clock and registers. The minimum delay for this design is primarily determined by its hold time. In comparison to a dynamic flip flop, this flip-flop has better noise immunity and does not require any refreshing.

Top Level Design

The top-level integration can be seen in Figure 2, with testbench results being shown in figure 1. The design includes all the integration requirements while also having a reset line

that sets the register back to 0. As seen by Figure 1 the design also operates with minimal delay. The main bottleneck is the adder as it is a ripple carry adder and must calculate 8 adds with no parallel computing. This is much more complicated than mux or register operations leaving the design to spend over 75% of its time during the adder stage. Choosing a more power hungry but speed efficient adder such as CLA or CSA would be a better choice for this design. The average power was 6083μW which was shown in the calculations section.

Conclusions and recommendations

As the critical path of our design works through two ripple carry CPL adders, these parts would have far greater delays than other parts in our implementation. A good way to reduce this delay would be to use carry look-ahead adders which implement separate logic to calculate the Couts and reduce the time taken for the Carry signal to propagate from the smallest bit forward. This would greatly reduce our propagation delay.

Calculations

gates = 36 * 8 for adder, 10*8*2 for Mux, 20*8 for register = 608 gates

Power = PDynamic + PStatic

Pd = C*V^2*f, C = 10f, V = 1V, f = 1Ghz

= 10μW * 608 = 6080μW

Assume leakage of 10na per gate

Ps = I*V, 10na * 608 gates = 3μW

Therefore, the total average power is 6083μW or 6.08mW

Appendix

Figure *1*: Top Level Test Bench Results

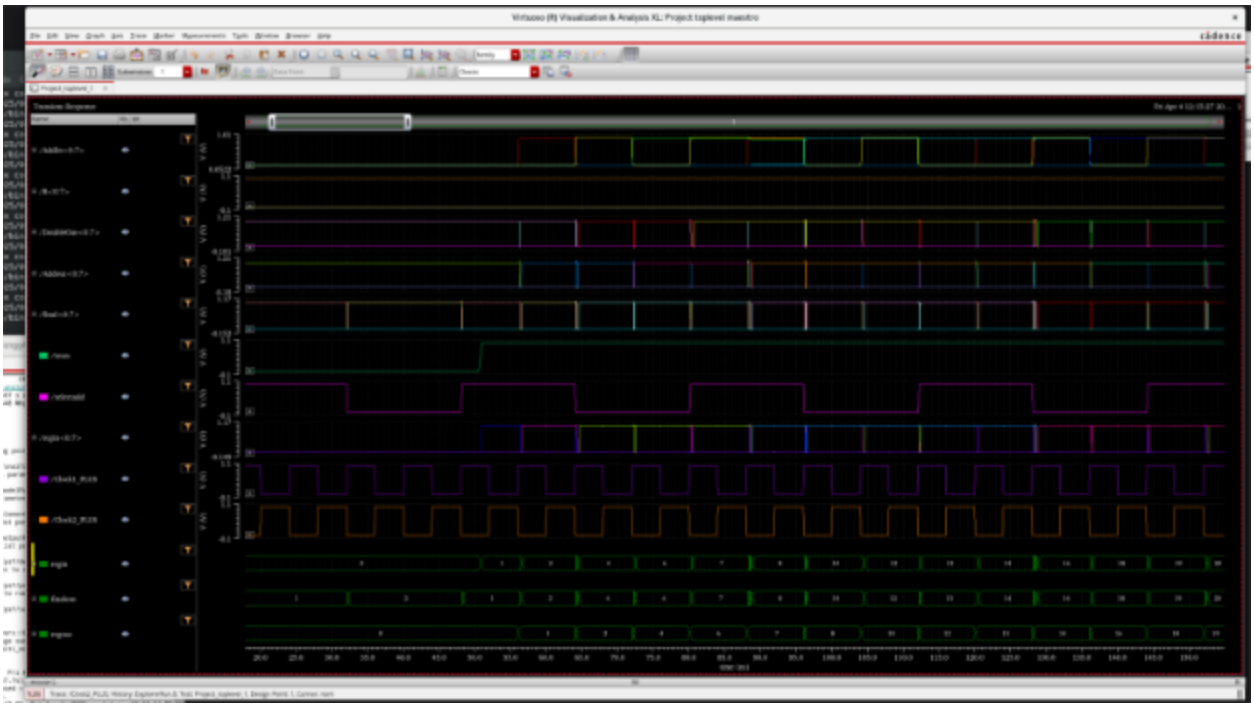

Figure 2: Top Level Schematic

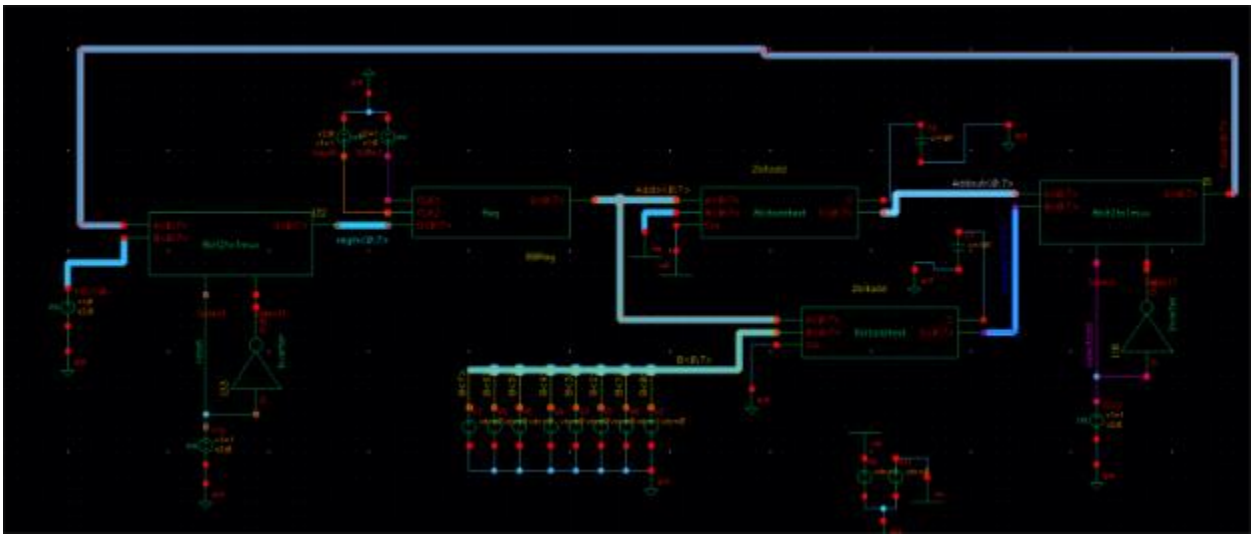

Figure 3: 8 Bit Mux Schematic

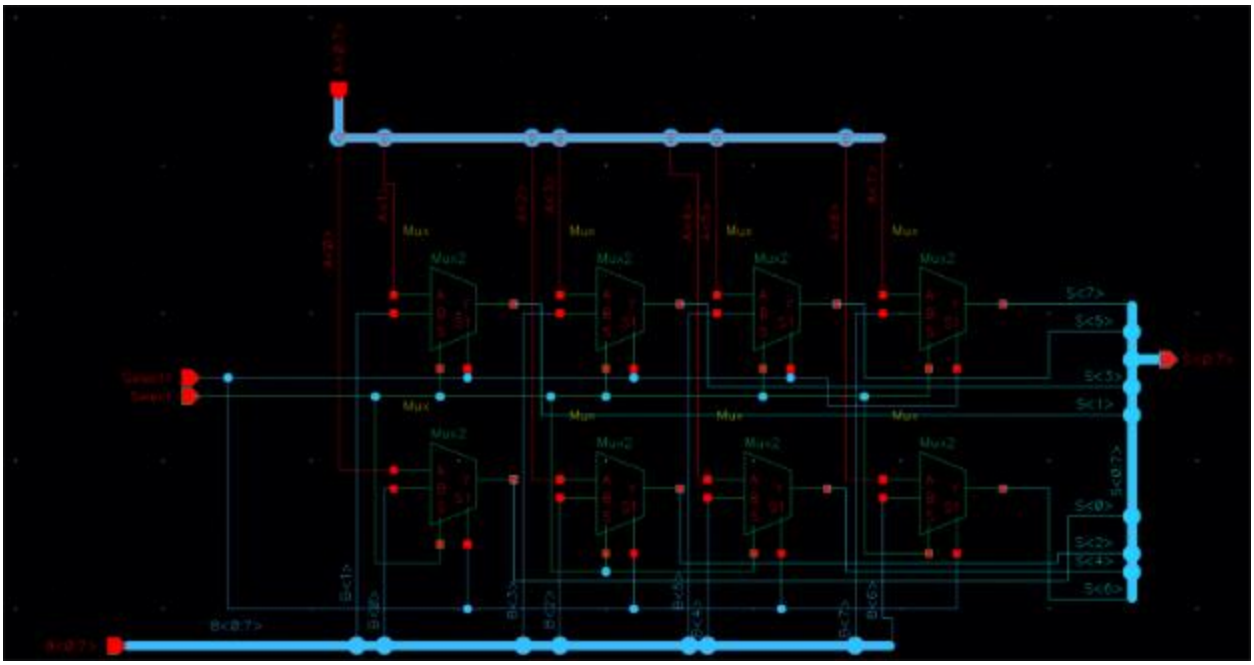

Figure 4: 1Bit Multiplexer Schematic

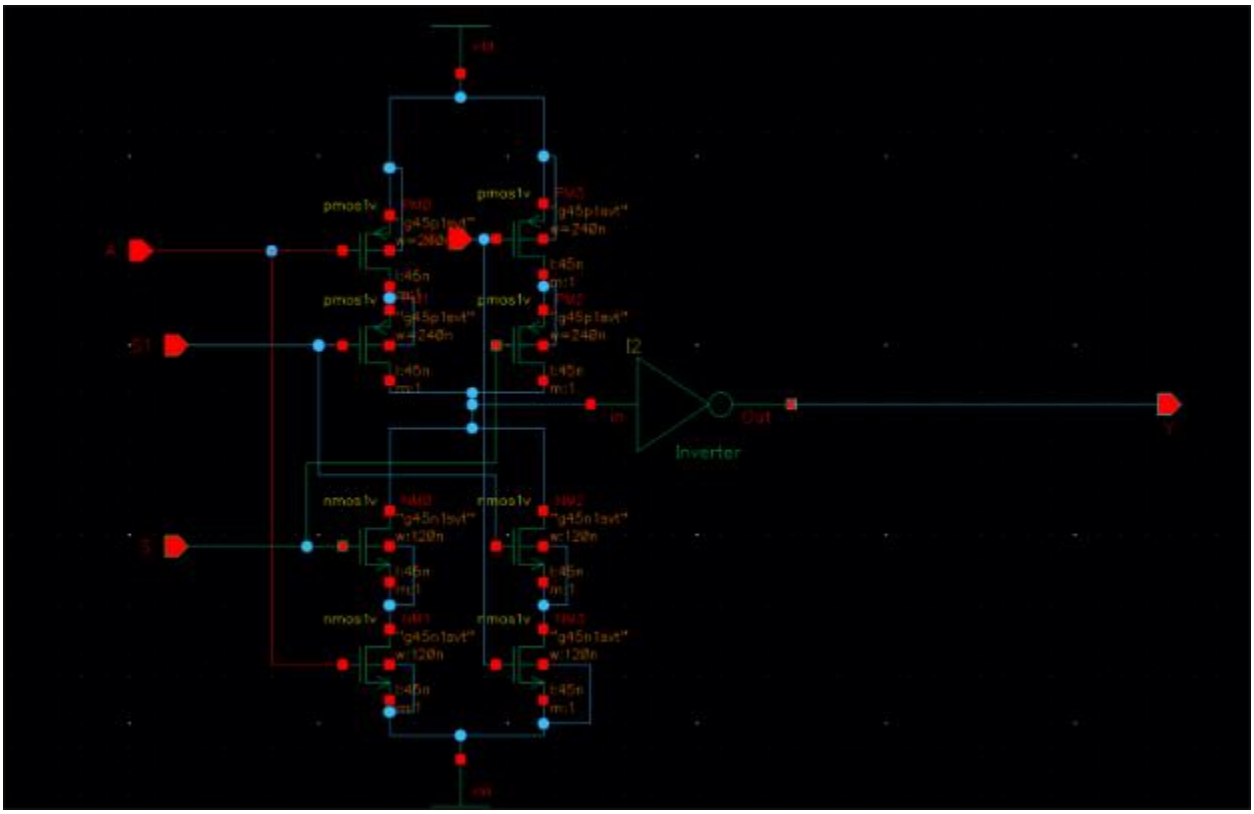

Figure 5: 8 Bit Register Schematic

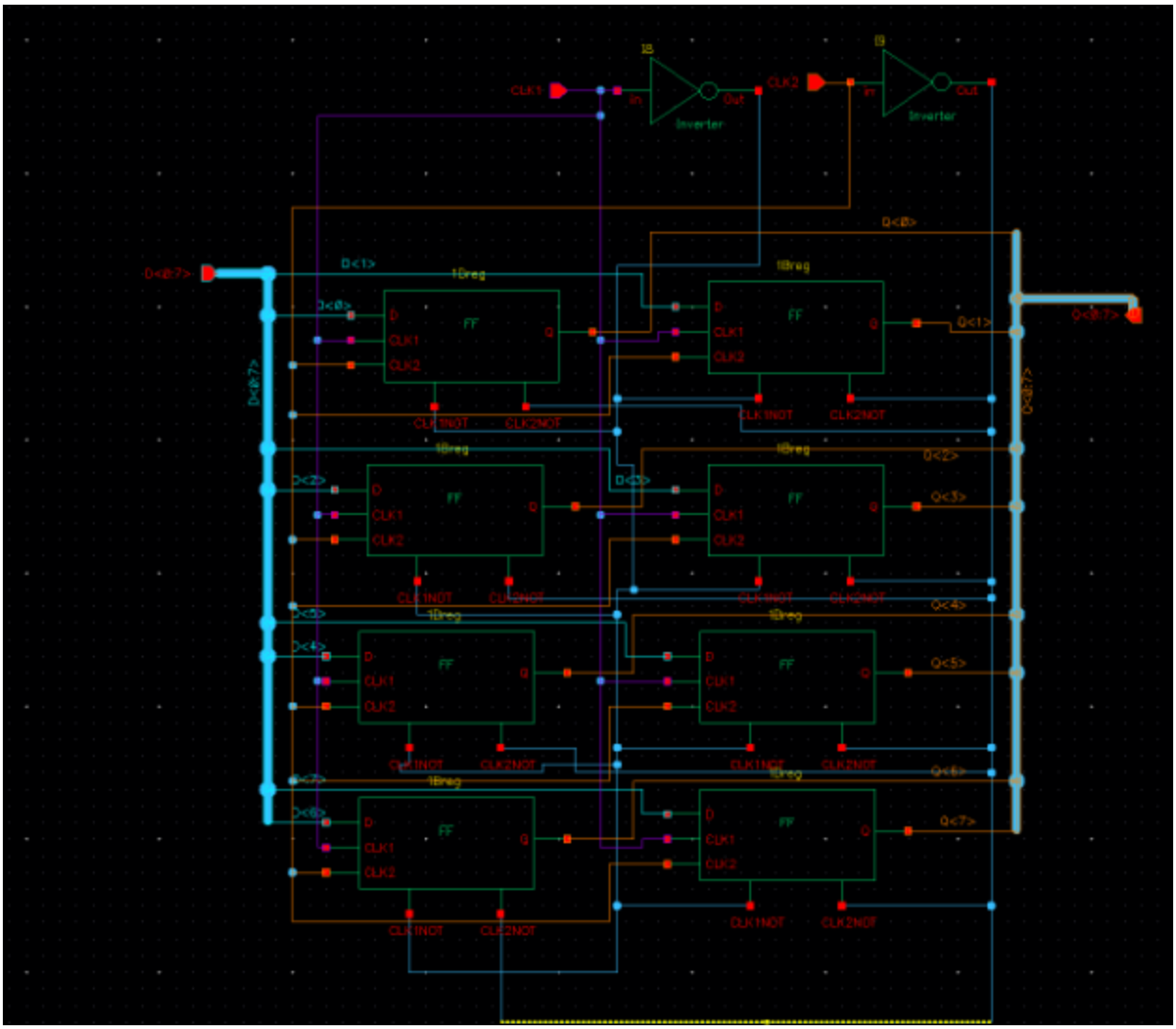

Figure 6: Single Bit Register Schematic

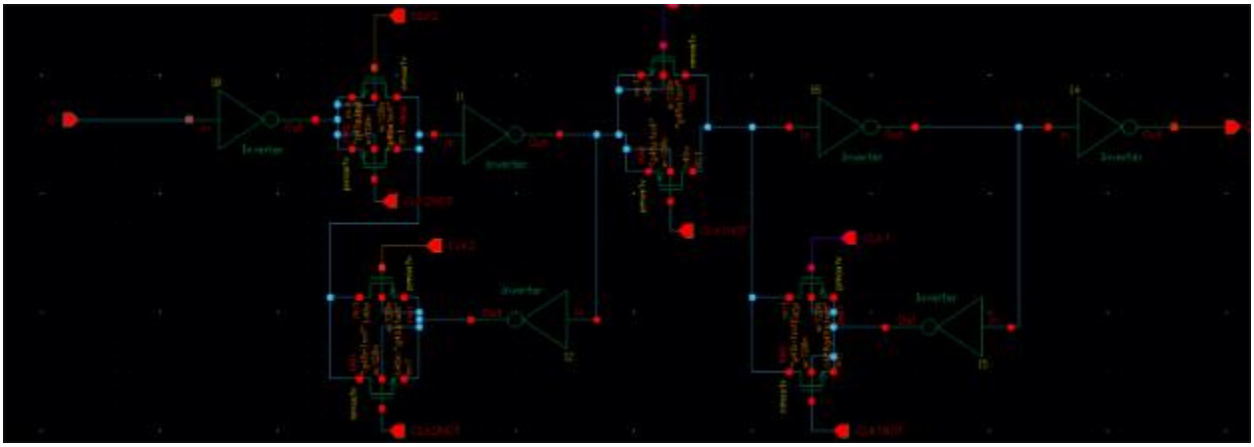

Figure 7: 8 Bit Adder Schematic

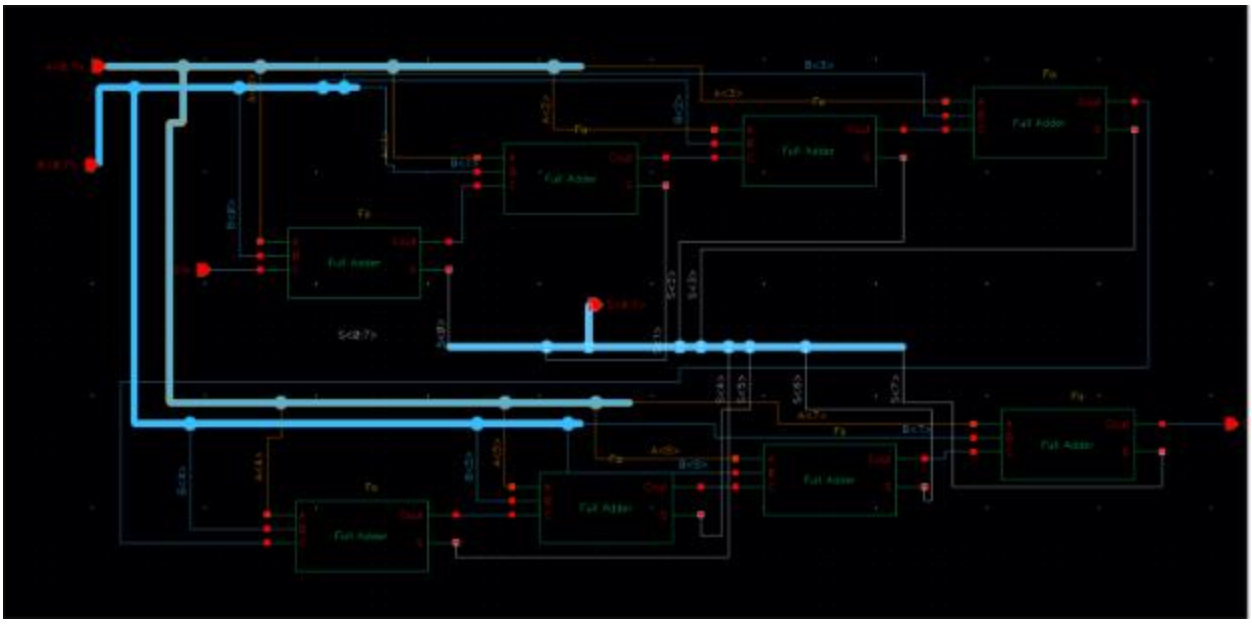

Figure 8: Single Bit Adder Schematic