SDC-92-371

SDC
SOLENOIDAL DETECTOR NOTES

FLOATING POINT ADDER TESTS

November 18, 1992

Srinivas Karuturi, Paschal Kibuule and Jheroen Dorenbosch
SSC Laboratory

FLOATING POINT ADDER

Srinivas Karuturi, Paschal Kibuule, Jheroen Dorenbosch
Physics Research Division
Superconducting Super Collider Laboratory

This report documents the tests performed at SSC Laboratory on a floating point adder, designed at Fermi National Laboratory [1]. The tests were carried out to evaluate the performance of this Adder. We also include a brief description of the test system used, and its setup.

Test System

We used the HP82000, which is a semiconductor test station capable of testing digital devices up to 100 Mhz. It has an edge placement accuracy of ± 350ps. The HP82000 has X windows based interactive software, through which the test engineer communicates with the tester.

Setup

1) The device under test (DUT) board was selected and the necessary wiring was done. To remove high frequency noise in the power supply to the DUT, we mounted decoupling capacitors between each power supply pin and ground.
2) We employed a Resistive Divider arrangement type of termination in which the DUT outputs were terminated to provide for impedance matching for the transmission line and the pin driver comparator electronics of the tester.
3) The pins were configured in the Pin Configuration window of the HP82000 software according to whether they were inputs, outputs, or terminated outputs.
4) This was followed by Level Setup where the low and high levels of input and output are defined, the Timing Setup where the clock frequency is specified and formats of signals and their delay relative to the clock are defined. For the input signals we used the Delay Non Return to Zero (DNRZ) data format. The clockI was set to Return to Zero (RZ) format with a 50% duty cycle. Finally the Vector Setup is done by loading the vector patterns appropriate for the test.

Software Developed

We developed C code to simulate the functionality of the ADDER . The ADDER is a floating point adder which adds two numbers in a non-standard format with an 8-bit Mantissa and 4-bit Exponent. The developed program first generates input vector data for the DUT based on the user specifications. The range of the mantissa and exponents, the starting and ending values can be specified by the user. The input vectors can also be specified as random. The program then takes these input vectors, performs the multiplica-

tion, and predicts the result in the proper format. The program converts both the input vectors and expected vectors from hex format into binary format. Vectors are organized in an output file such that each line represents the states of all the ADDER inputs and outputs at a given time. The propagation delay through the chip was properly taken into account.

We also developed shell script to load input vectors and expected vectors into the vector setup file of the HP82000. This program first converts the test and expected vectors from ASCII to HP format. It then loads the input vectors using the command "DRVD" and expected vectors using the command "EXPD" in the vector setup file. From the HP82000 control window we execute an in-built function "acquire data" . The ADDER chip adds the list of numbers and the output is placed in the "state list" file which can be compared with the expected vectors. Any errors are highlighted automatically and can be seen in the state list window.

The shell script can be used to load both drive data and expected data of any chip into the vector setup file of the HP82000. The data can be either in binary or in hex format.

Test Results

We evaluated Rise and Fall times, minimum pulse widths of the clock, Mantissa A and B, timing response to temperature, power supply variation, and power consumption of chip. Also we verified Clocked mode speed under the same condition the ADDER would be used in a system, and examined worst case throughput times and found worst case input patterns.

We started testing the chip initially with 256 vectors and the chip seemed to work at the specified frequency of 50 Mhz in the Clocked mode and 66 Mhz in the FallThru mode. We also loaded 60K vectors (not random) and repeated the test. The ADDER was working at 50 Mhz in the Clocked mode. But as we increased the complexity of input patterns and the number of test vectors(to around 100K) the chip did not work at the specified frequencies in both the Clocked and Fall Thru mode. We then took random patterns of 4K vectors and found the following results:

The maximum speed at which the ADDER worked for the selected vectors was 43.48 Mhz(23.0 ns) in the Clocked mode, with ClockI being at 43.48 Mhz and patterns that changed at 43.48 Mhz. We learned that the inputs and outputs are Clocked on the trailing edge of the clockI. Below are some of the results obtained when the chip was functioning at this frequency in the Clocked mode:

1) The Setup Time of the inputs is -2.20 ns, measured w.r.t the trailing edge of the ClockI (fig 1).
2) The Hold Time of the inputs is 2.75 ns, measured w.r.t the trailing edge of the ClockI..
3) The ClockIn to ClockUndelayed out total propagation delay is 14.7 ns and was measured w.r.t the leading edge of the ClockI.
4) The ClockIn to ClockDelayed out total propagation delay is 17.7 ns. Hence the internal delay of the clock fanout tree is 3.0 ns

5) The propagation delay of the outputs (both the Mantissa and Exponent) is 21.1 ns and was measured w.r.t the leading edge of the ClockI. The output appears after two cycles hence the actual propagation delay is 21.1+ 46 = 67.1 ns.
6) The Data Hold Time of the outputs is 19.2 ns, measured w.r.t the leading edge of ClockI.
7) The power consumption of the chip is 535mW.
8) As to how the ADDER may be used in a system environment can be explained using figure 2.

The maximum speed at which the ADDER worked for the selected vectors(4K of random vectors) was 30.30 Mhz(33.0 ns) in the FallThru mode. The correct output appears at about 60 ns after the inputs change and is valid for less than 1 ns only.

Some of the other values which were investigated are as shown below.

1) The minimum pulse width of the ClockI is around 3.2 ~ 3.5 ns
2) The minimum input pulse width of Mantissa A and Mantissa B is around 3.0 ns with valid output.
3) The chip functions correctly when the supply voltage is varied from 4.7 to 5.5 Volts.
4) The rise time of the outputs is around 1.5 ns and the fall time is 1.6 ns.
5) The chip worked in the temperature range of 10 °C to 50°C and failed beyond 51°C.
6) We have taken a sample of 4K vectors and found the failing vectors in the Clocked mode (with clockI at 45.45 Mhz). After identifying the failing patterns we put them in a different vector file and ran the tests again. Interestingly some of them passed which may suggest that they were failing earlier because they have been preceded by different vectors. But some of them still failed. We have included a sample of the failing vectors and their expected outputs along with internal register values and sent them to R. Yarema et al. for consideration.

In conclusion, we have tested the floating point adder chip. The results obtained indicate that some work still has to be done in order to make this chip work at the required speed. In a separate study we have shown that chips with equal functionality can be designed using a Silicon Compiler. We expect such chips to run well above 100 Mhz. Floating point adder can also be implemented using Xilinx or other programmable devices.

References:

1) M. Larwill, J. Hoff, C. Needles, " Adder Tests"

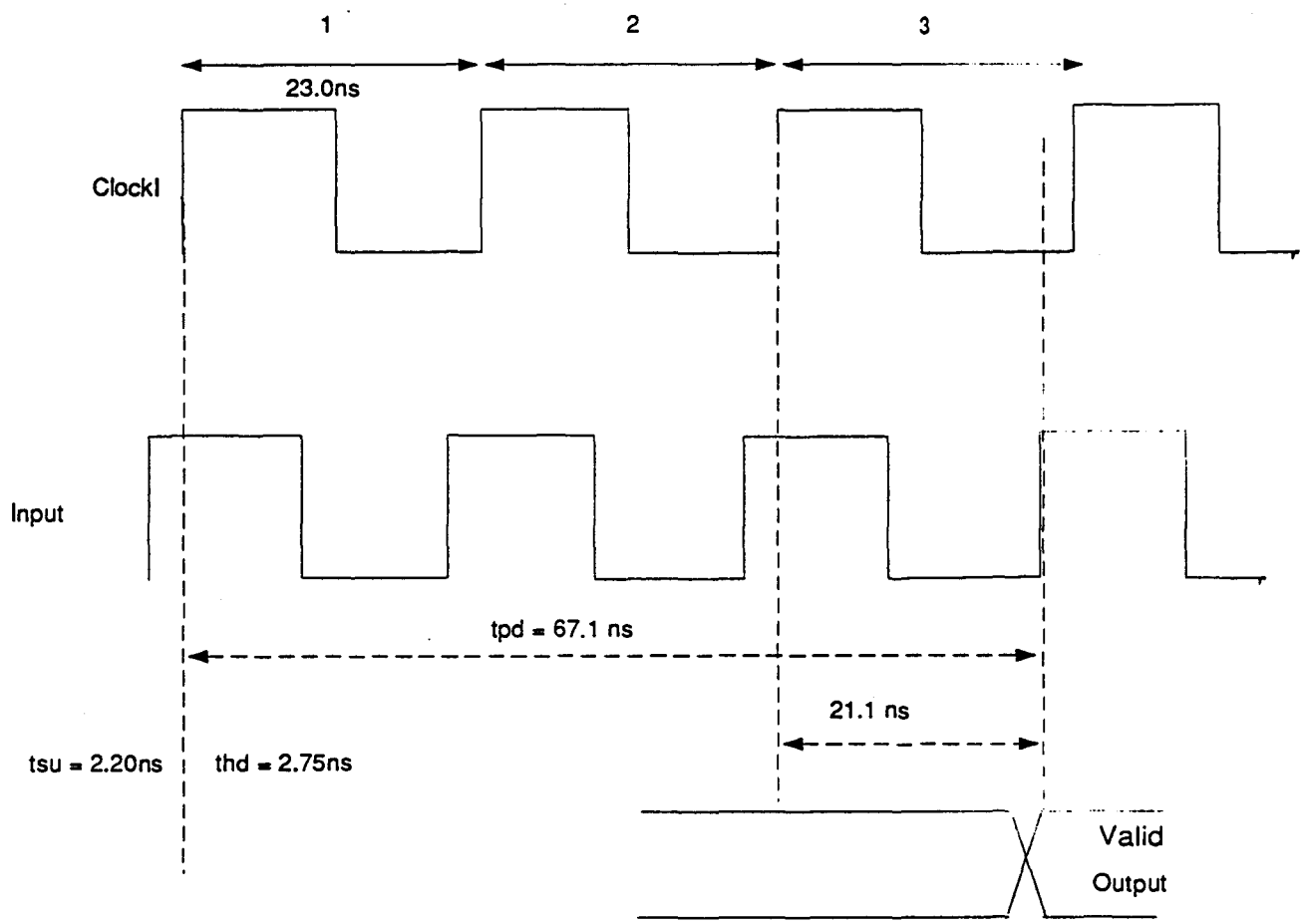

Figure 1 shows propagation delay, setup and hold times

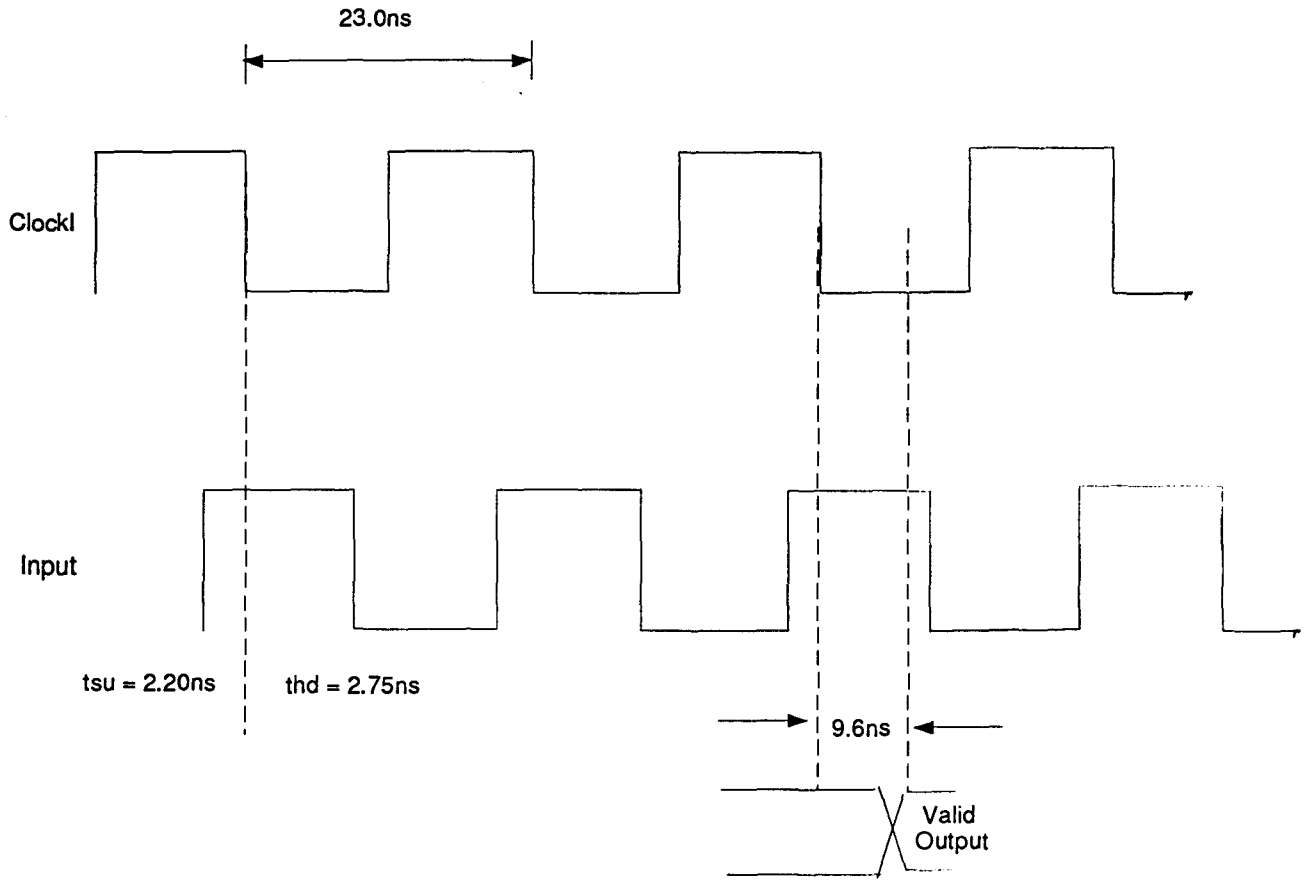

Figure 2

The Propagation Delay from the falling edge of the ClockI is 9.6ns as shown. So minimum time. required to process an input signal is 9.6ns + 2.20ns + 2.75ns + ~ 3.0ns Delay (i.e delay between chips or between boards) which comes out to be around 17.55ns.