PROJECT 9

EFFECT OF DATA CORRELATION ON FIR FILTER

Submitted by
Deepaknath Tandur K.
Sumit Ahuja

ABSTRACT:

This project deals with effect of data correlation on various implementation of FIR Filter (Finite Impulse Response Filter). Here FIR filter is implemented in RTL netlist style using VHDL. Two implementation methods have been considered, one of them involves no resource sharing and other involves sharing of resources. The implementation is then synthesized down to the gate level and power consumption is estimated using Synopsys Power compiler. In this project an attempt is taken to study the effect of data correlation on different implementations using various set of meaningful input data streams. It is observed that for parallel implementation, data correlation highly affects the power consumption (especially dynamic power consumption). But with the resource sharing based implementation the effect of data correlation with power consumption is less.

I. INTRODUCTION

Reducing power consumption is a necessary challenge for today's ASIC designers. Silicon technology advances have made it possible to pack millions of transistors, switching at high clock speeds on a single chip. While these advances bring unprecedented performance to electronic products, they pose difficult power dissipation and distribution problems. These problems must be addressed, because consumers demand longer battery life in addition to lower cost in computers, battery-operated systems, medical devices, telecommunications equipment and many high-volume consumer products.

Application related to Digital Signal Processing in which a lot of computation is involved has got a great importance in portable devices such as mobile phones. Power efficient design for DSP functionalities is the demand of the industry today. In this project FIR filter, which is very important component of many DSP applications, is implemented and effect of data correlation is studied on different type of design styles.

Next part of the report discusses the design of a simple FIR filter and different implementation styles. In part III data correlation and its effect are discussed. Part IV explains the results obtained and concludes the report.

II. DESIGN OF FIR FILTER

Among the various possible implementations of FIR filter we have chosen direct form realization. Six tap FIR filter has been considered. The figure below depicts the parallel implementation of FIR filter:

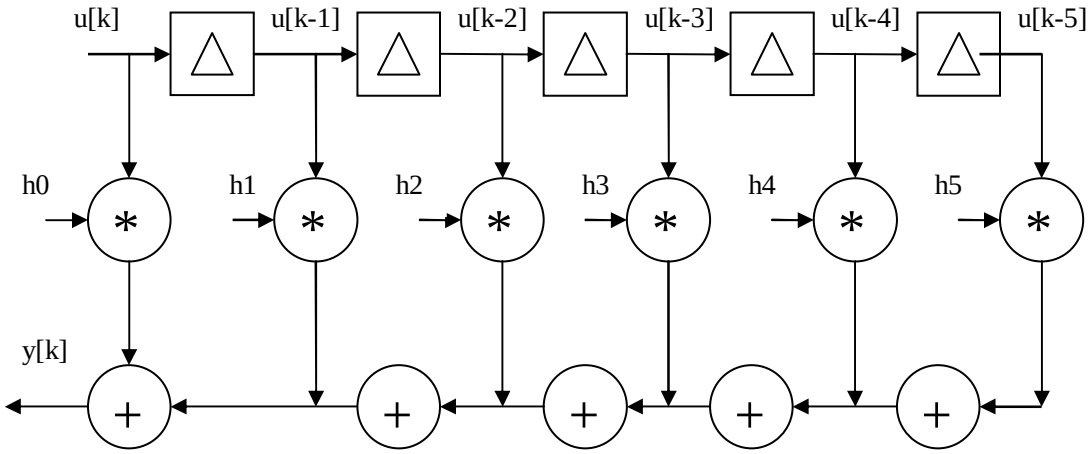

Fig1: Direct Realization of FIR filter with unshared resources.

For RTL level netlist generation we have used structural multiplier and structural ripple carry adder. For non resource shared FIR (parallel implementation of FIR), six multipliers and five adders are instantiated as components for performing multiplication and addition respectively.

The second type of implementation involves the sharing of resources. In this implementation we use only one instance of multiplier and adder. We have used pipeline style of implementation where in each clock cycle both the multiplier as well as the adder is used. Because of resource sharing the latency is high. The FIR takes around 7 clock cycles to give the output y(k), which is depicted in figure.

In both implementations FIR works on 8 bits. The figure below shows the implementation style:

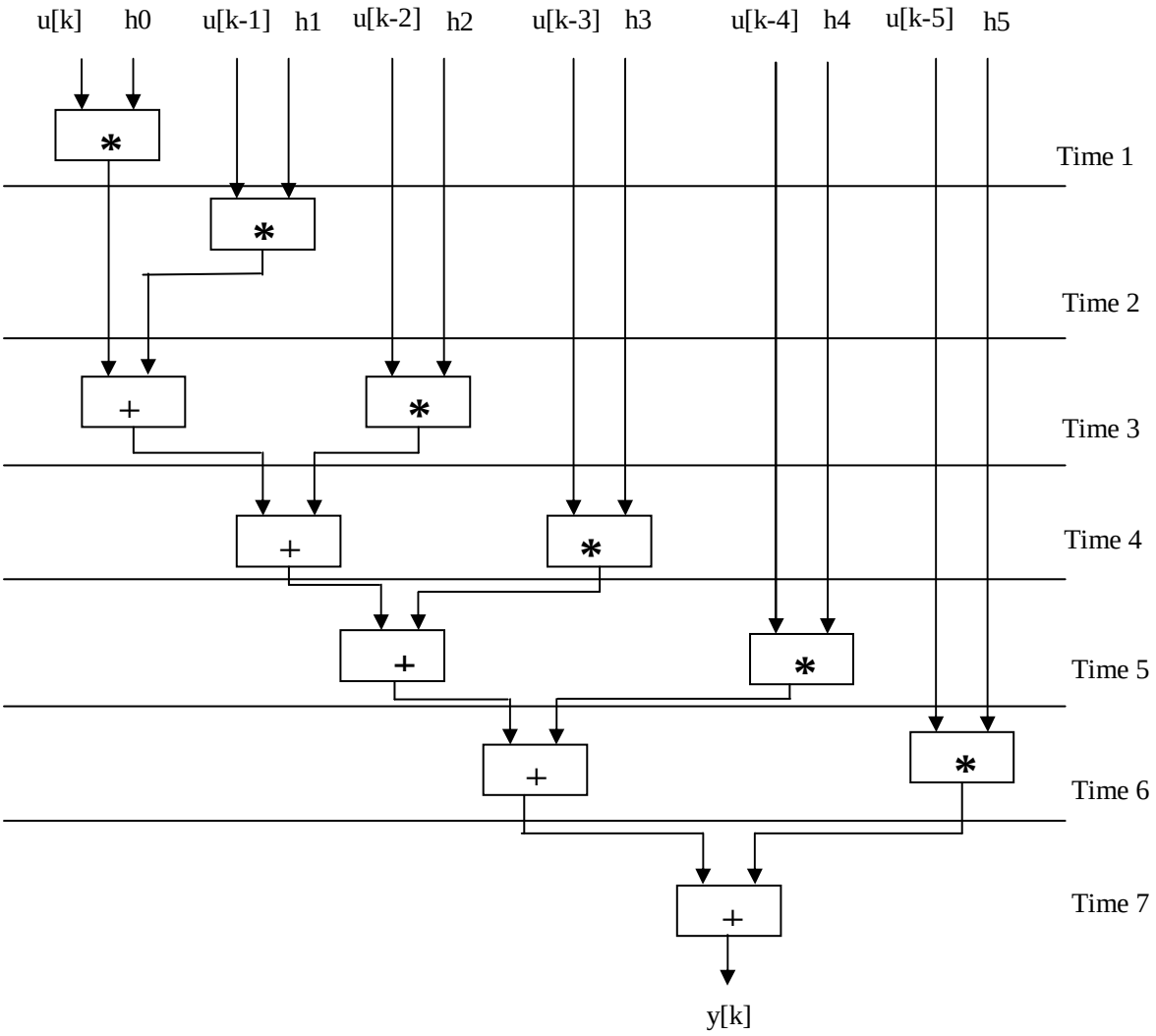

Fig2: Direct Realization of FIR filter with shared resources.

III. DATA CORRELATION.

The power consumption in execution units depends on switching activity of operands, that is, data correlations between successive inputs. It has been shown that more positively correlated input data stream consumes less power. If we assign operations to execution units through scheduling and binding in such a way that the input data correlations increase, power consumption of execution units can be reduced.

From the structure of FIR with shared resource based implementation, we can observe that the inputs of the multiplier are multiplexed from different sources, so for each output of the 6 tap FIR (in our case), the same multiplier takes six different inputs. Hence the effect of data correlation to the multiplier input is destroyed. Because of which even for highly correlated and uncorrelated input data pattern the change in power dissipation is relatively very small.

In case of unshared resource based (parallel) implementation, input to each multiplier changes only once for every output. Thus the data correlation is highly observable i.e. for highly correlated data the switching activity will be low and vice versa. Because of this, the activity factor highly depends on correlation in input. Thus power dissipation which is proportional to the activity factor gets equally affected.
Five different types of data pattern are taken in test bench:

Highly Correlated Input pattern
In this type of pattern there is only one bit change in each successive input. For example: 00000000, 00000001, 00000011 and so on.

Correlated Data Pattern
In this type of pattern, binary inputs increment by one. For example:
00000000, 00000001, 00000010, 00000011 and so on.

Mid-correlated Data Pattern
In this type of pattern, two bit change is applied to successive inputs. For example:
00000000, 00000011, 00001111, 00111111, 00001111 and so on.

Uncorrelated Data Pattern
In this type of pattern, four bit change is applied to successive inputs. For example:
00000000, 00001111, 11111111, 00001111 and so on.

Highly Uncorrelated Data Pattern
In this type of pattern, change in all the bits is applied to the successive inputs. For example: 00000000, 11111111, 00000000 and so on.

IV. RESULTS AND CONCLUSIONS

For both types of implementation different input patterns (as discussed in the section above) are applied and the power consumption is estimated for gate level synthesis. Simulation of Parallel implementation for a particular input is shown in fig 3.

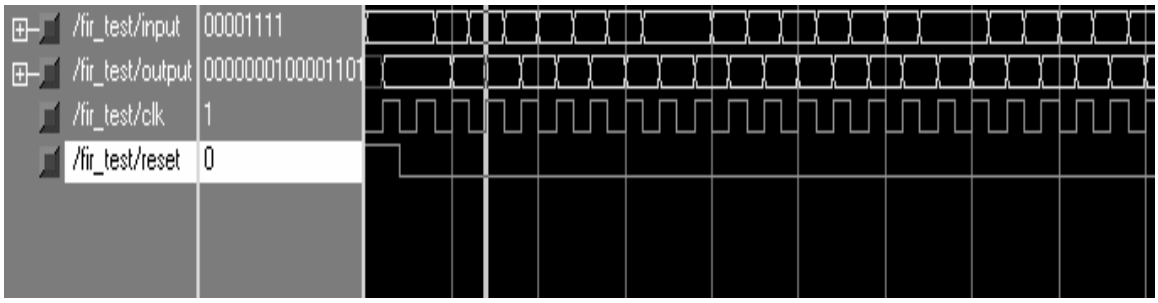

Fig 3: Simulation result for FIR parallel implementation.

The graphs in fig 4 and fig 5 show the effect of input pattern on power consumption for both implementations.

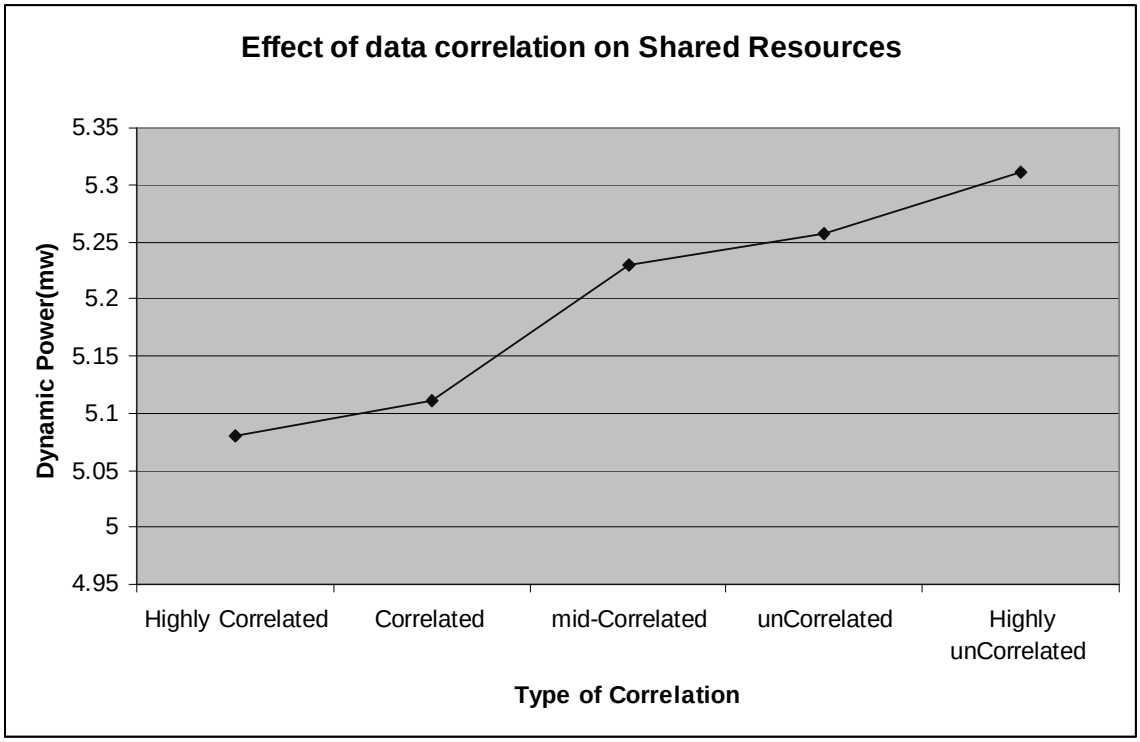

Fig 4: Effect of data correlation on Shared Resources.

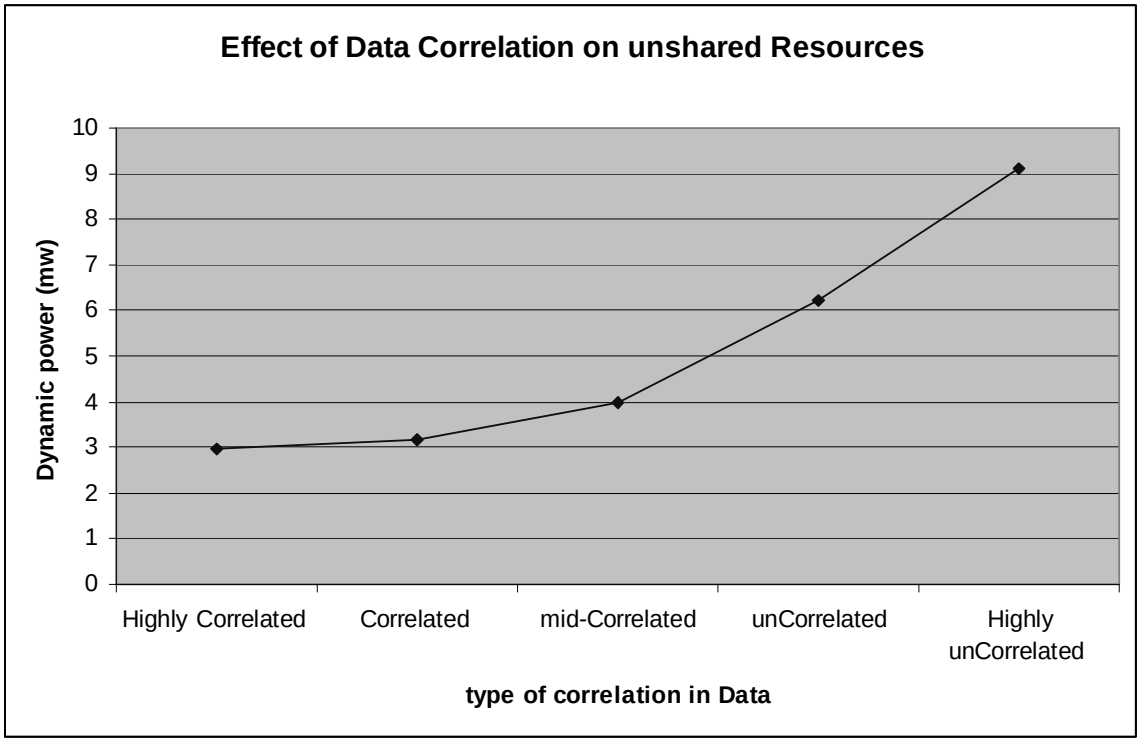

Fig 5: Effect of Data correlation on Parallel Realization.

As it can be seen from the graphs, the effect of data correlation on Dynamic power for unshared resources is very high. Dynamic Power of lowest correlation is approximately 3 times higher than that for the highest correlation of data input. Thus it proves that the Switching activity for highly correlated data is low and vice versa.

While the effect of data correlation on shared resource is not very high. It shows that data correlation is destroyed because of this type of implementation.

This case study shows that effect of data correlation is very considerable for low power devices. It shows the importance of implementation style with respect to data correlation and power dissipation as in some cases these effects can be minimized. A trade off for power versus area, speed versus power etc. can be done by changing the implementation style (e.g. sharing more resources) by considering the effect of different data correlations.

References:

1. S. Chandrakasan, R. W. Brodersen, ``Minimizing Power Consumption in Digital CMOS Circuits," Proceedings of the IEEE, Vol. 83, No. 4, pp. 498-523, April 1995.
2. http://www.dspguru.com/sw/opendsp/alglib.htm
3. Discrete-Time Signal Processing (2nd Edition) by Alan V. Oppenheim, Ronald W. Schafer , John R. Buck .
4. Synopsys Design Compiler Manual.
5. Synopsys Power Compiler Manual.
6. DSP Class notes for ALaRI by Prof. Marc Moonen.