GRC Task - 1988.001 - Targeting Multi-Core Clock Performance Gains: Vertically Integrated Adaptation and Prototyping

I. Summary and Key Research Accomplishments

The key research accomplished for this task is broadly divided into two parts (as shown in Figure 1) and the key research accomplishments for each part is stated below

(1) In the first part (system tuning), a low cost post-manufacturing testing and speed tuning methodology is proposed in a multi-processor system. The goal of this research is to develop a methodology that allows the "safe" speed of each core in a large CMP to be determined under the assumption that some speed defects and design bugs are likely to escape conventional delay testing procedures. The processors form pairs and operate in redundant mode and compare execution results (signatures) at regular intervals. At the end of the tuning process, maximum speed for reliable operation in each processor core is determined. The self-tuning process provides additional confidence for reliable operation over existing post-manufacturing techniques in the presence of difficult-to-model and difficult-to-detect bugs. To account for application dependent bugs and failures due to wear-out, the proposed speed tuning methodology is designed to be applied on-field during real application operation without any system downtime or/and rollback recovery support.[1]

Key Accomplishments:

- A 2-dimensional accelerated collaborative binary search algorithm is proposed which significantly reduces the total speed testing and tuning time (by 10.34%) in a multi-core system of processors.
- A fault simulator is designed, to simulate application dependant speed faults in a MIPS pipeline and calculate the total time (in clock cycles) taken to tune a system for reliable operational speed.
- A publication in Asian Test Symposium, 2011[1] and SRC Techcon 2011[2]

(2) The second part (micro-architecture tuning) of the research argues the need to redesign every processor pipeline in a multi-processor system with a timing variation tolerant adaptive pipeline design. We propose a novel globally synchronous and locally asynchronous adaptive pipeline design, that automatically adjusts to timing variations, thus reducing/eliminating the need for design guard-banding. Each pipeline stage is equipped with an activity completion sensor that determines switching activity completion in the respective pipeline stage with a very high probability of correctness. The switching activity completion detection is used to trigger handoff of data from one pipeline stage to another. The pipeline is equipped with low overhead back-up error resilient circuits to handle incorrect (but infrequent) completion sensing and continue forward progress. Such a scheme enables time lending between pipeline stages, thus allowing correct pipeline computations at a lower voltage/higher clock frequency as compared to fully synchronous designs.[3-4]

Key Accomplishments:

- A novel architecture for globally synchronous, locally self-timed pipeline processing is proposed and its power and performance benefits are studied in a multi-stage prototype pipeline design.
- A probabilistic switching activity completion sensing methodology and sensor design is developed for the first time. It includes (1) A Minimum Gate Sensing Algorithm to reduce the sensor overhead (2) pseudo- NMOS based activity sensor design, which generates the activity completion signal. This method is the first of its kind and provides an alternate direction to look into adaptive systems than it has been in the past.
- Development of a backup error detection and correction mechanism that ensures correct operation of the pipeline and continuous forward progress in the event when the pipeline does not function correctly
- The power and performance benefits (considering the overheads for completion sensing and error detection) are demonstrated using synthesis, simulation tools and custom timing

simulators. Work is in progress (nearing completion) on a prototype design of an adaptive FIR filter using a Xilinx Vertex-6 FPGA and PCB board.

- Alternate techniques of adaptation using completion sensing (for error prediction) is being explored (as summarized in Section IV). Simulation and hardware results will be published and reported to SRC and NSF by February 2014.
- Publication in DAC 2013[3], VLSI 2014[4] and 2 transactions (to be submitted and uploaded in SRC website by March 2014).

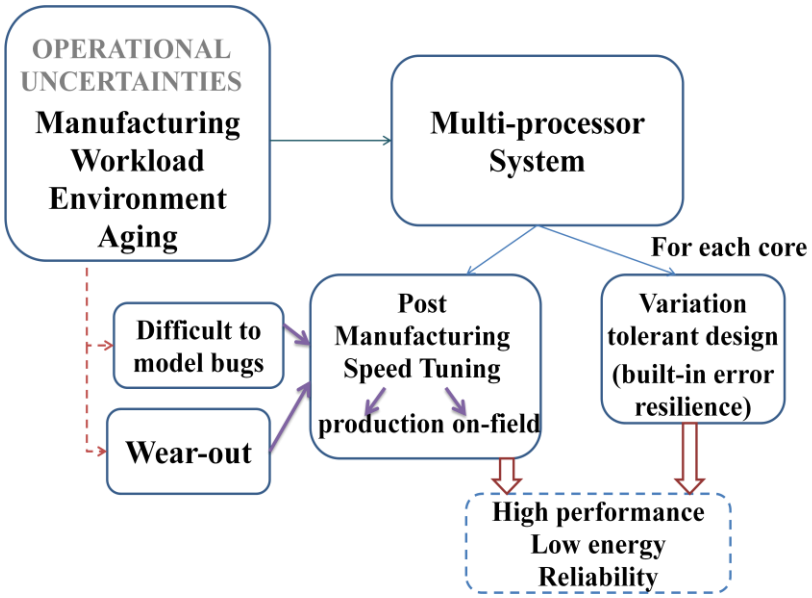


Figure 1: Summary of research

In the rest of the document, the research techniques, methodology and results are summarized.

II. Distributed Comparison Test Driven Multiprocessor Speed-Tuning: Targeting Performance Gains Under Extreme Process Variations[1]

The system level overview of the proposed approach is shown in Figure 2. Following are the key features of the proposed distributed speed-tuning methodology

1) Initially, pairs of processors are formed. Speed testing and tuning is performed using an iterative test-tune-test methodology which involves running identical test instructions or identical program threads of a real-time application on the pair of cores and comparing signatures generated at inserted program checkpoints.

2) Signatures at checkpoints of test application instructions or checkpointed application programs are compared and marked trusted/not trusted at the frequency of operation at which testing is performed.

3) At any point during instruction execution, the execution results of one of the processors in the pair can be trusted. Hence we do not need to compare execution results with a stored golden reference making it completely a self-tune based methodology.

4) The distributed iterative algorithm converges in O (log F_p) steps where 'F_p' is the number of discrete clock speeds possible. It is independent of the number of cores in the system. The error checking technique is shown to have high fault coverage (see simulation results)

5) Aggressively scaled technologies undergo gradual device wear-out due to gate-oxide breakdown, hot-carrier injection, Negative Bias Temperature Instability (NBTI) etc which appear as timing failures at the onset of degradation. The proposed technique can be applied online/in the field periodically to prolong system lifetime at the cost of core clock speed.

6) The proposed technique requires minimum hardware overhead and uses some of the inbuilt features (e.g multiple supply/frequency operation) of current processors. The kernel/operating system needs modification to support application instruction/thread duplication during test-tune mode.

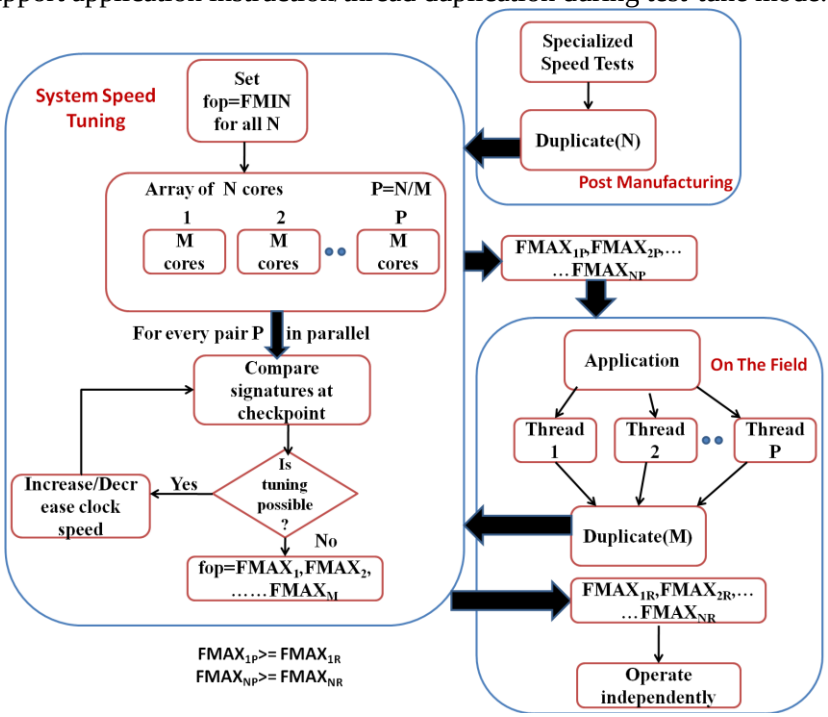
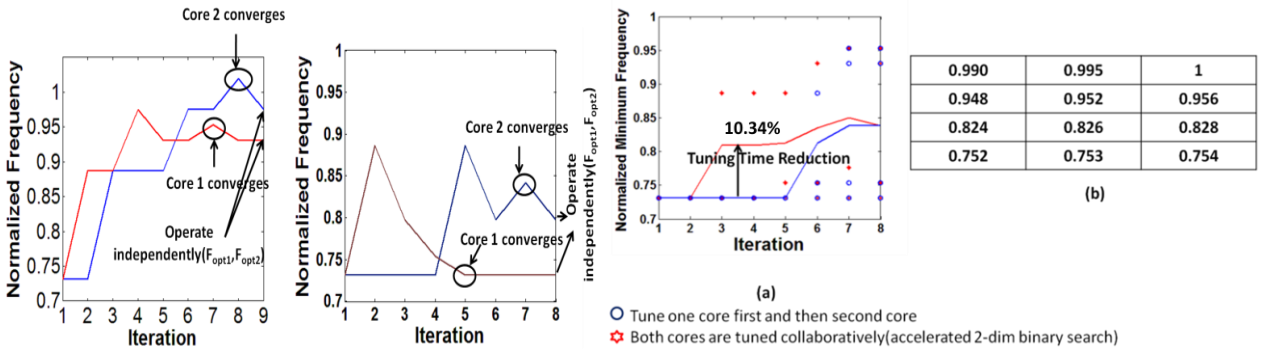


Figure 2: System level overview of the proposed approach and the tuning procedure applied to a multiprocessor array of N core

i. Collaborative 2-D accelerated binary search algorithm

Tuning algorithm selection is done considering the following (1) the iterative test-tune procedure should converge as fast as possible. If algorithm converges faster, it decreases test-tune time during post-silicon validation. On the field its impact on application throughput and performance is reduced (2) During tuning, select ′M′ (no. of cores to be coupled) such that throughput impact is minimum on application while ensuring checking procedure does not fail (3) While tuning the clock frequency of ′M′ processors in each pair, the total time to execute 'I' instructions is limited by the processor operating at the lowest clock frequency among the 'M' processors. This minimum clock frequency should be as high as possible (4) Fail-safe operation of the application and OS on field while tuning.

The collaborative two-dimensional accelerated binary search algorithm iterations for two different pairs is shown in Figure 3(a). Only one core changes its frequency in the next iteration thus keeping one core trusted always during tuning. After convergence, the cores operate independently with speed F_{opt1} and F_{opt2} where F_{opt1} and F_{opt2} are the frequency points as supported by the PLL and closest to but lesser than FMAX1 and FMAX2 respectively thus attaining maximum clock performance.[1]



0.990	0.995	1
0.948	0.952	0.956
0.824	0.826	0.828
0.752	0.753	0.754

(b)

Figure 3: (a) Accelerated 2-dim binary search iteration steps for (a) Normalized FMAX2= 0.990 and FMAX1=0.948, (b) FMAX2= 0.824 and FMAX1=0.752 (b) Benefits of Accelerated 2-dim binary search

III. Timing Variation Adaptive Pipeline Design: Using Probabilistic Activity Completion Sensing With Backup Error Resilience[2-3]

The block diagram of the proposed design is represented in Figure 4. Each pipeline stage (P2,P3,P4) is equipped with a sensor, which monitors switching activity of few selected logic gates in each pipeline stage. Each sensor generates an "activity completion signal" with a high probability of correctness, after activity in the respective pipeline stage has completely ceased. Data is handed off using Time Lending Flip Flops (TLFF) from one pipeline stage to the next when the neighboring pipeline stages have generated the "activity completion signal" (as controlled by each AND gate in Figure 1(a)). An incorrect activity completion is handled by built-in error detection and correction circuits. Such a design with the ability to hand-off data upon switching activity completion, relaxes the hard bound imposed by fully synchronous clock driven designs. This gives intermediate pipeline stages the flexibility to adjust to timing variations (as indicated in Figure 4(b)) in the pipeline in every clock cycle. In order for the pipeline to interact with the outside world, the first and last stages of the pipeline are synchronized with the global clock, henceforth denoted as "virtual clock".

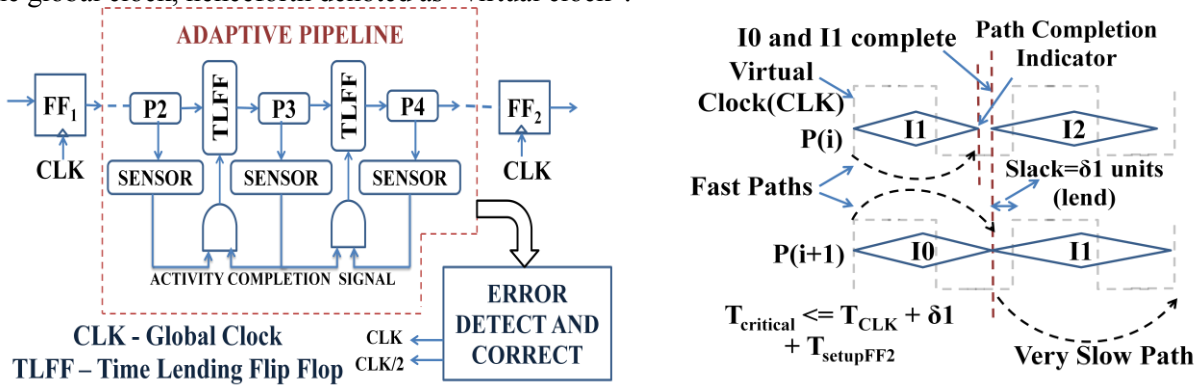


Figure 4: (a) Block diagram of the proposed adaptive design (b) : Timing diagram showing time lending across successive clock cycles. T_{VSP}- Path delay of a very slow path, $T_{setupFF2}$ - Setup time of FF2

i. Minimum Gate Sensing Algorithm

The problem is formulated as described below:

Given: A collection of switching gates with switching times for $'M'$ stochastically generated input vector transitions $\{T\}$, which forms the gate switching log file.

Goal: From the gate switching log file, form a minimum gate set $\{G_{opt}\}$ such that for every input transition in $\{T\}$, there exists at least one gate in $\{G_{opt}\}$ which switches during consecutive **'Δ'** intervals until switching ceases.

Solution:

(1) Initialize $\{G_{opt}\} = \emptyset$. Let $G = \{G_1, G_2 \dots\dots\dots, G_n\}$ be the set of all logic gates in the combinational logic circuit

(2) Form a matrix A, where each column represents logic gates in $\{G\}$ and each row represents a particular **'Δ'** interval for an input transition in $\{T\}$. A value of 1 in any matrix element indicates that the corresponding gate in that column switched during the corresponding **'Δ'** interval of the input transition. For e.g a 1 along the row $T2_1$ indicates the gate switches for input transition T2 in the first **'Δ'** interval (refer to Figure 4).

(3) Find the gate which switches maximum number of times among all $\{T\}$ for all 'Δ' intervals by column-wise addition of the matrix elements of A to form a single row vector. *By selecting the gate which switches most number of times, the overhead is minimized while working towards satisfying Criteria 1 for all every input vector in $\{T\}$ (greedy search).*

(4) Add the selected gate to $\{G_{opt}\}$.

(5) Eliminate all the rows of A for which the selected gate switches and form a matrix A' with remaining rows. Assign A'=A

(6) Repeat Step 3 to 5 until all the rows of the matrix is eliminated. *By eliminating all rows, we guarantee that the goal as stated above is satisfied for each input transition vector in {T}.*

Figure 5(a) pictorially explains Steps 2 to 5.

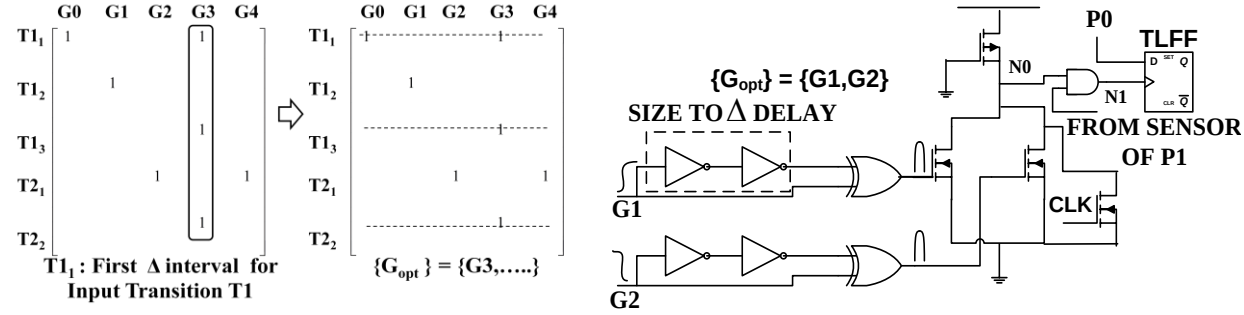


Figure 5: (a) Gate selection in Minimum Gate Sensing Algorithm (b) Activity completion sensor design

ii. Benefits of proposed adaptive design over progressive over-clocked design (Razor based) and fully synchronous designs

Figure 6(a) shows that for the same error rate, the proposed design can support higher average clock speed (higher throughput), in applications with low power requirements, the proposed design has a potential to support lower average operating voltage across a range of workloads than progressively over-clocked designs(Razor based designs).

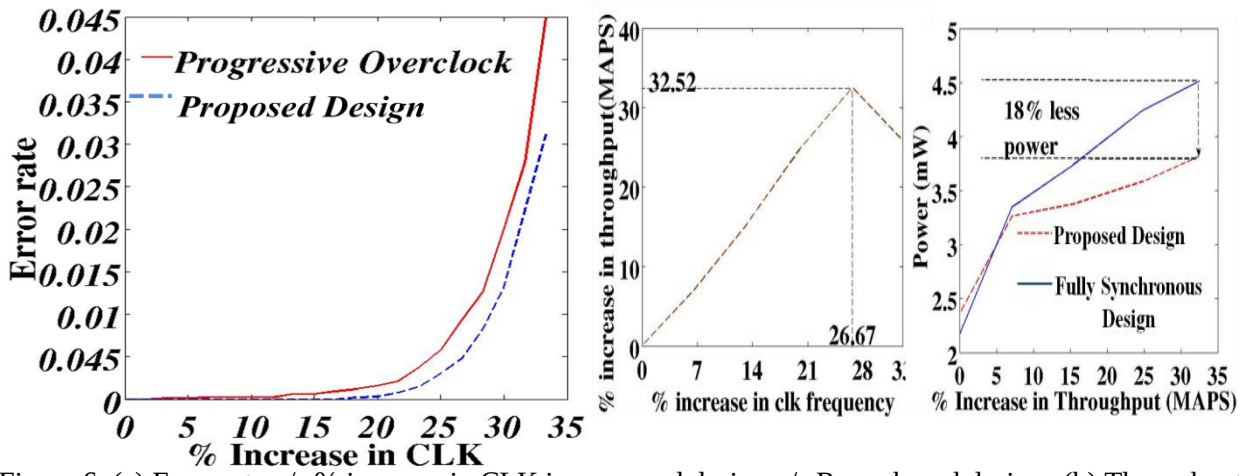


Figure 6: (a) Error rate v/s % increase in CLK in proposed design v/s Razor based designs (b) Throughput and power benefits in proposed design over a fully synchronous design

In order to study the benefits of the proposed adaptive design, we compare it with a fully synchronous design with worst-case margins (Figure 6(b)). To increase the throughput of a fully synchronous design,

both supply voltage and frequency are increased simultaneously such that all the critical paths in the circuit meet the timing constraint imposed by the synchronous clock. In the proposed adaptive design, the clock frequency is increased while keeping the supply voltage constant (1.1V in NCSU45nm PDK). Errors are detected and corrected by incorporating error detection and correction circuits as explained in Section 5. Figure 6(b) plots % increase in throughput v/s % increase in clock frequency for the proposed design. MAPS stands for Multiply Additions per Second.

IV. Conclusion and Future Direction

Multi-level adaptive methods in multi-processor systems have been demonstrated and shown to give significant performance/power benefits (with reliable operation) in the presence of extreme process variations, workload variations, environment variations and aging. Accelerated collaborative muti-core tuning algorithm can reduce total time during post manufacturing speed binning. Our research work shows that adaptive systems built using completion sensing is an excellent replacement for the traditional fully synchronous systems. The completion sensing methodology by selective sampling of gate output nodes is the first of its kind and provides an alternate direction to look into adaptive systems in future.
Along this direction, in the near future using the completion sensing methodology we plan to look into error prediction schemes for a fully synchronous system as opposed to the described error detection scheme.

i. Error prediction using completion sensing

Referring to Figure 7, selective output nodes from the pipeline stages of a fully synchronous pipeline are sampled into a sensor using the similar technique proposed in Section III. The sensor generates the completion signal. The completion signal (instead of triggering hand-off) is used by a Pulse Detection Circuits (PDC) to determine the time interval Δ at which completion occurs over successive clock cycles in every pipeline stage. This information is encoded and used as a feedback mechanism to increase/decrease the clock frequency or supply voltage. Such a mechanism can predict trends in variations of workload, environment which might potentially cause the critical path to fail in future clock cycles.

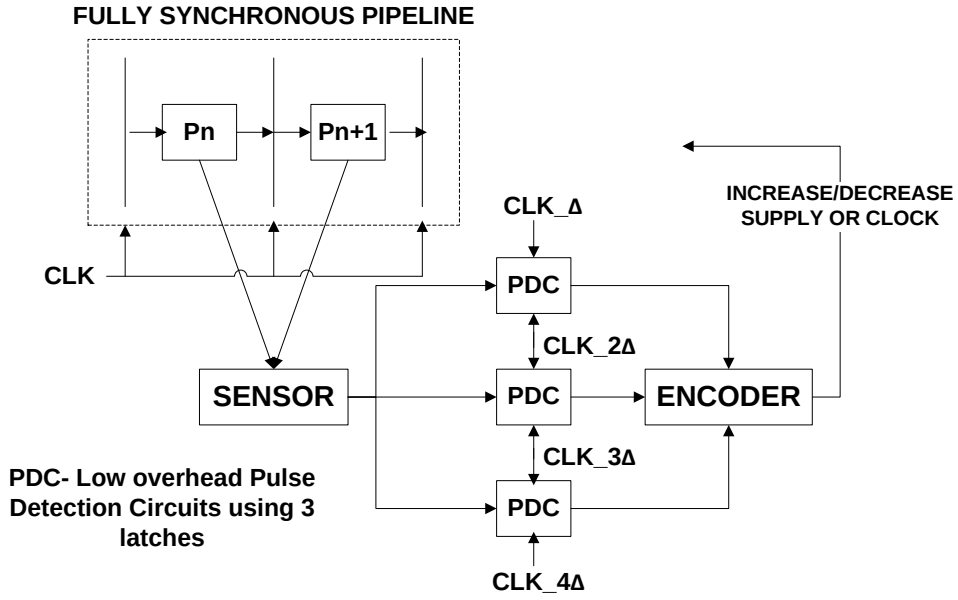


Figure 7: Error prediction using completion sensing detectors

The benefits of this technique will be quantified and summarized in another report which will be uploaded into the SRC website by March 2014. Hardware prototype using FPGA and PCB board (as described in

subsection ii of Section IV) will demonstrate the benefits of such a technique in a pipeline FIR filter design.

ii. FPGA and PCB prototype to demonstrate the benefits of completion sensing in a pipeline FIR filter

We are in the process of building a PCB prototype of the completion sensor and use a Xilinx Vertex-6 FPGA to program a FIR pipeline implementation. The basic structure of the prototype design is shown in Figure 8. A pipeline FIR filter is programmed into a Xilinx Vertex 6 FPGA using array multipliers and Ripple Carry Adders (RCA). Selective internal gate output nodes of the adder and multiplier combinational logic are routed into the output of the FPGA using I/O pins. The selection of gates from adder and multiplier whose output nodes is monitored is done as per the "Minimum Gate Sensing Algorithm" as described in Section III.

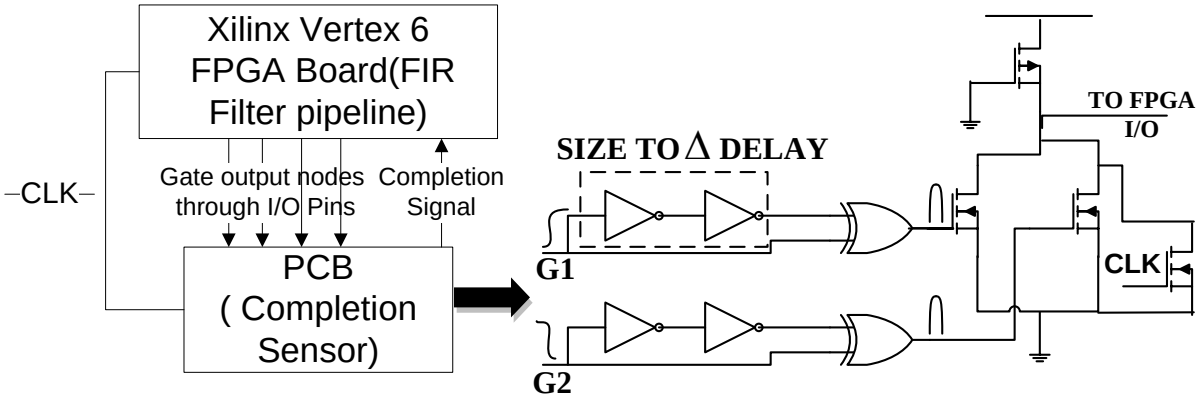


Figure 8: Prototype design of adaptive design using Xilinx Vertex 6 FPGA and Customized Completion sensor design using a PCB

Hardware Setup

The FPGA is connected to the PCB board using the Fast Mezzanine Card (FMC) connectors present on the FPGA board. Following components are designed and included in the PCB (1) male interface for the FMC connector (2) Sensor design using standard off the shelf ICs of inverters and XOR gates and PMOS and NMOS transistors. The Δ delay matched with the block delays on the FPGA. Both the FPGA Board and PCB are synchronized using a common clock supplied using a clock generator. The completion signal is generated on the PCB using design shown in Figure 8 and routed back to the FPGA which is used to hand-off data from the adder to the multiplier pipeline stage.

Validation

A pipelined 8-bit multiplier followed by a 16-bit adder is designed in verilog and programmed into the FPGA. This is the first step of the overall design. The pipeline design is verified at different clock frequencies and the results are shown in Figure 9. The inputs to the pipeline adder multiplier are applied using a Block RAM. The outputs are also captured into the Block RAM which is designed to perform multiple reads and writes in one clock cycle using multiple I/O ports. As shown in Figure 9(b), in a two stage pipeline, the output Z= X*Y+C appears two clock cycles after the application of inputs X and Y. C is constant (FIR filter co-efficient).

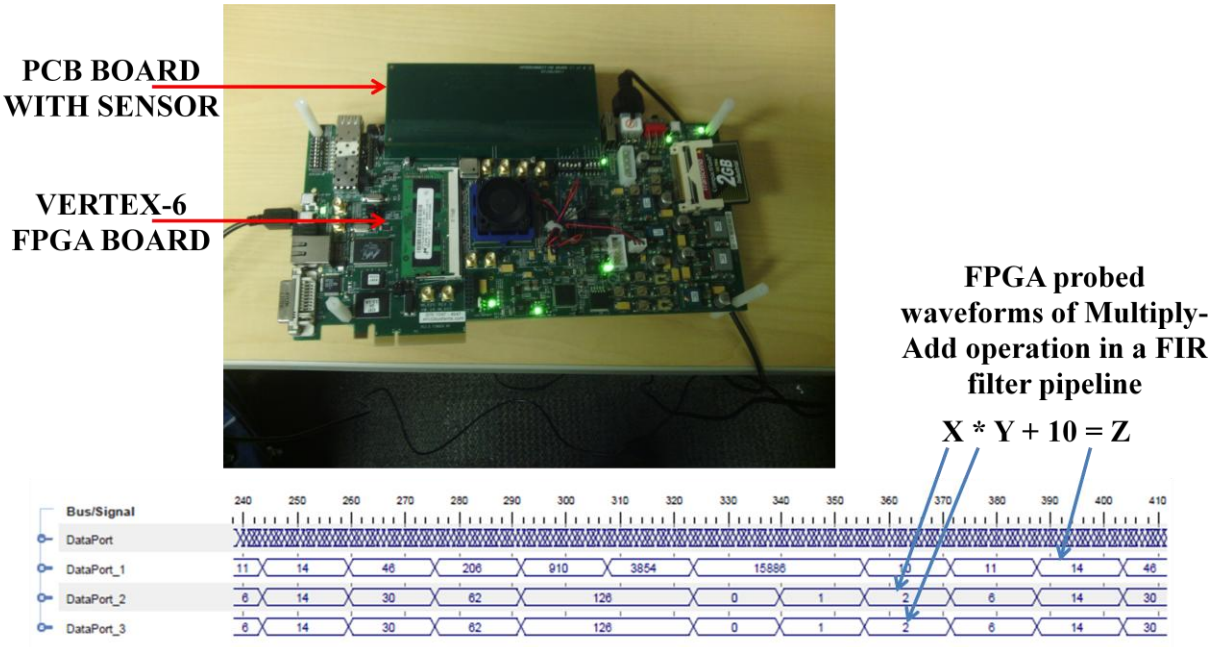


Figure 9: (a) Figure shows FPGA plugged into the PCB using FMC connectors (b) Waveforms of Chipscope (software debugger) showing verified pipeline design of 8-bit multiplier- 16-bit adder programmed into the FPGA

Experiments (to be done and reported by March 2014)

The following experiments will be performed to demonstrate the benefits of the proposed adaptive completion sensor based pipeline design

(1) Verify the pipeline using a reference maximum clock frequency as operated in fully synchronous manner and as a globally synchronous and locally synchronous manner.
(2) Calculate the power overhead of the sensor.
(3) Increase the clock frequency and monitor error rate with a without time sharing using the completion driven pipeline
(4) Quantify the power and performance benefits considering overhead of completion sensor.

V. Publications

[1] Natarajan, J., Wells, J., Chatterjee, A., & Singh, A. (2011, November). Distributed Comparison Test Driven Multiprocessor Speed-Tuning: Targeting Performance Gains under Extreme Process Variations. In *Test Symposium (ATS), 2011 20th Asian* (pp. 154-160). IEEE.

[2] Natarajan, J., Wells, J., Chatterjee, A., & Singh, A. (2011, November). Distributed Comparison Test Driven Multiprocessor Speed-Tuning: Targeting Performance Gains under Extreme Process Variations, in SRC Techcon 2011

[3]Natarajan.J et. al,"Timing Variation Tolerant Real Time Adaptive Pipelines using Wave Completion Sensing", presented at DAC 2013 as part of WIP Session

[4] Natarajan. J et. al, "Timing Variation Adaptive Pipeline Design: Using Probabilistic Activity Completion Sensing With Backup Error Resilience", to appear in VLSI Design 2014

[5] Two journals to be submitted to IEEE Transactions and will be uploaded to SRC website and reported to NSF by March 2014