Self-timed circuitry for global clocking

Scott Fairbanks
Cambridge University
Scott.Fairbanks@cl.cam.ac.uk

Simon Moore
Cambridge University
Simon.Moore@cl.cam.ac.uk

Abstract

We present an apparatus used to distribute a timing reference or clock across the extent of a digital system. Self-timed circuitry both generates and distributes a clock signal, while using less power and less skew compared to a clock tree. HSpice simulations in a 180nm CMOS process comparing the ***Distributed Clock Generator*** *presented in this paper and an H-tree clock distribution system, each clocking a 16mm × 16mm area suggests a 30% power savings. Also worst case skew was reduced from 27ps to 2ps while using a clock period equivalent to 9 FO4 gates.*

1. Introduction

1.1. Related Work

This paper extends the thread of research that investigates the timing relationship of the handshaking control signals generated by self-timed FIFO circuitry as it passes data tokens.

Ebergen, Fairbanks, and Sutherland [4] showed that data tokens in self-timed linear FIFOs can act as if they exert forces on each other. They showed that the forces are sometimes repellent and sometimes attractive. The *Charlie effect* was the name given to the repellent force.

The Charlie effect describes how a second token is repelled as it nears a first token that is before it in the FIFO. They identified the source of the Charlie effect as the increased delay of the logic gates in the FIFO control when the inputs to the logic gate change near the same time. The increased logic gate delay results in the FIFO stage taking longer to move the second token forward. The increased move delay pushes the second token back away from the first token.

Winstanley, Garivier and Greenstreet [12] placed data tokens into a ring FIFO and designed novel circuitry that could vary the relative strength of the attractive and repelling forces, alternately causing the tokens to bunch and spread in the FIFO. When they increased the strength of the repelling force they noticed that the tokens spread evenly through the FIFO.

In [5] we designed a simple Micropipeline-style [11] FIFO control that amplified and exploited the Charlie effect, so that the tokens in a FIFO spread throughout it without the aid of an external control. We formulated the relationship between the FIFO's occupancy and the phase relationship of the signals on the handshake control wires. We also outlined a method of distributing the FIFO over an area, loading the FIFO with some number of tokens, and using the resulting high frequency signals with precise phase relationships for various timing applications.

We then described how this Micropipeline style FIFO control, while using digital handshake protocols to control the ordering of tokens, was acting as an analog feedback circuit. The delay of the many FIFO control elements varied depending on the relative arrival and departure times of data tokens. This effect ensured that the timing signals on the handshake wires had a phase relationship that could be predicted with pico-second accuracy even in the face of transistor mismatch. Suggested applications ranged from high frequency sampling to generating control signals for domino logic.

1.2. The Task

This paper once again uses self-timed circuitry to achieve high precision timing. Rather than address the task of sampling at high frequency or controlling domino logic as in our previous paper, the concern of this work is global clocking. The former tasks involve generating timing many signals with precise phase relationships. Global clocking in its simplest form requires a timing standard of a single phase distributed to many widely separated nodes. Both problems require high precision timing and both solutions actively correct timing errors using the token spreading feedback mechanism. However, the particular demands of clocking a large digital system leads us to a different and novel circuit topology than that presented in our earlier paper.

Three concerns motivate the drive for a new self-timed circuit topology beyond that presented in [5] for use as a global clock distribution system. First, power consumption is of elevated concern in clock distribution. Second, the geometry of the clock distribution system is of ever increasing performance. A clock distribution system capable of clocking a billion-plus transistor system at frequencies greater than a giga-hertz must have a topology that integrates well with the power distribution apparatus and functional units that compose the rest of the digital system. Third, design time and design complexity is a factor. The practice of balancing a growing number of electrical paths between a clock source and the state holding elements is increasingly untenable.

Power and skew The task in clock distribution is ensuring that periodic events occur in many location at the same instant in time. The prevalent clock tree solution prescribes many electrically equivalent paths from a single source to the many locations. Because this is a passive or open-loop solution, the longer the paths are from the source, the more skewed in time are the many clock events at their destinations due to inevitable fabrication mismatches.

Designs often employ two remedies to control the skew between the clock events at the different destinations. The first remedy makes the electrical distance between the source and the leaves shorter using wider conductors and larger amplifiers. This reduces latency and the time over which events can drift when branching from a single node. The second remedy is to make low resistance paths between the many places needing the clock signal, in essence, ‘shorting together’ the nodes requiring the clock signal. A clock grid employs this strategy.

The enabling commodity in both solutions is power. Both require more hardware, which presents capacitance that must be charged and discharged in each cycle. Reducing power and reducing skew in the clock distribution apparatus are largely the same problem. If power is available, skew can be reduced. Unfortunately the efforts of supplying power to and removing heat from the chip themselves present formidable challenges. A compelling clock distribution solution should suffer less skew for a given power budget.

Regularity and geometry State of the art VLSI chips are billion-plus transistor systems. Their complexity necessitates the integration of the complete interconnect architecture [13]. The power distribution system, clock distribution apparatus, and signal interconnect utilize the same interconnect stack. The design of one system must consider and balance the needs of the others.

If we consider a microprocessor with a view to its geometry, the most salient detail is its ’rectangularity,’ see Figure 1.

Signal routes are confined to run parallel to the X and Y axis of the chip. The functional units are typically of a rectangular shape. The power is typically distributed in a rectangular mesh whether sourced through peripheral pads or through a flip-chip bonding array. Power and ground rails are available at regular intervals along the X and Y axis of the chip. The VLSI fabrication process lends itself to rectangular topologies. Fabrication equipment steps along one axis of the silicon, laying down masks and etching in the appropriate place.

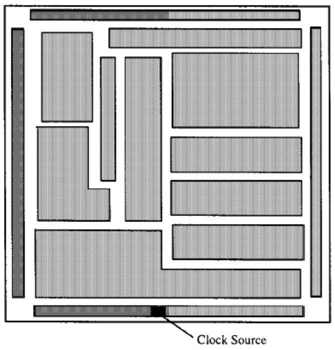


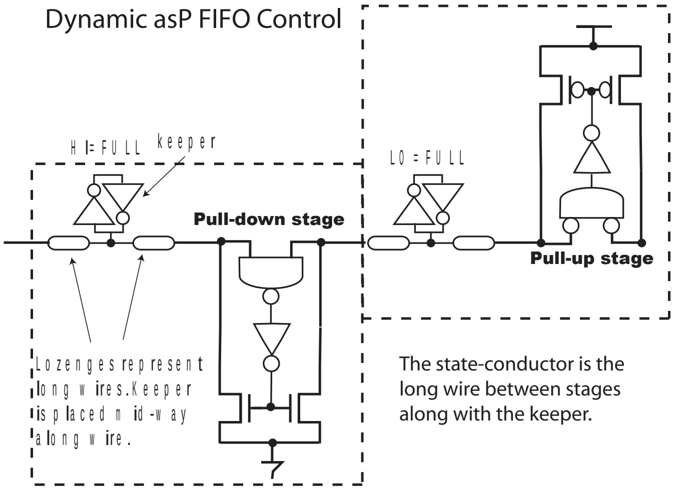
Figure 1. Image of Typical Microprocessor Chip

Continued progress in the art of VLSI relies upon the ability to confine materials, voltages, and currents to specific places and paths. Regular and congruent structures facilitate this goal. The clock tree structure that is employed to amplify a signal from a single source and distribute itself over a large two dimensional area is incongruent with the power grid and the rectangular shapes of the other structures. A clock distribution solution streamlined with the existing geometry of the functional and power distribution systems facilitates our ability to trace current paths, isolate noisy from sensitive signals, and allocate physical resources.

Design time The costs spent in tools, design time, and computer cycles achieving design closure must be considered. The incongruous relationship of a clock tree topology and the other systems on the chip requires making difficult concessions with respect to routing, sizing, and placement. Each alteration in the place or route of circuitry could require a complete clock tree re-balance because it is very difficult not to confound routes for the clock signal with the routes of other signals.

2. Pulse control for global clock

Figure 2 shows two stages of **Dynamic asP** FIFO control. **Dynamic asP** has been previously disclosed only in the form of a patent [7]. The overall control uses two types of alternating FIFO control stages. The potential on the state conductor that connects stages is changed to alert a control element to the presence of a data token. Each FIFO control element sets the potential on the up and down stream state conductors simultaneously to the same value when a token is moved forward by a FIFO control element. The presence of a token in a stage is signaled by setting the potential on the state conductor to a certain potential. The stage=FULL signal is assigned to opposite polarities in alternating stages. **Dynamic asP** FIFO control also needs latches that become transparent on opposite polarities in alternating stages.



Lozenges represent long wires. Keeper is placed mid-way along wire.

The state-conductor is the long wire between stages along with the keeper.

Figure 2. One Pull-up and One Pull-down stage of Dynamic asP FIFO control

The alternating nature of **Dynamic asP** makes it a less than ideal FIFO control. But **Dynamic asP** is attractive for the purposes of generating and distributing a clock signal. When this FIFO control is loaded with a number of tokens within some range, then the tokens spread apart because of the Charlie effect. If the number of tokens is equivalent to half the number of stages, then the signals on the state conductors that connect the control elements will oscillate in phase. These signals can be buffered and used as clock signals.

We believe that **Dynamic asP** FIFO is the minimal FIFO control for producing single phase handshaking signals. Furthermore the handshaking protocol is enacted on a single communication wire. These two factors suggest that this FIFO control could distribute a single phase signal most economically with respect to power and hardware.

2.1. Proposal

A ripple FIFO ring built from **Dynamic asP** FIFO control, is routed between the supplies of the power distribution grid as shown in Figure 3. Absent noise, transistor mismatch, and start-up transients, all state conductors connecting stages oscillate in phase if the FIFO is loaded with a number of tokens equivalent to half the number of FIFO stages. The mid-point of the state conductors replace the leaves of a clock tree.

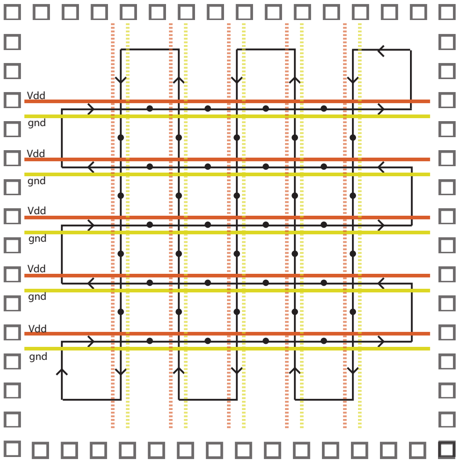


Figure 3. Ripple FIFO routed to form a clock grid

2.2. Problem

If the digital system being clocked occupied a 16mm × 16mm chip and each control element separated from its neighbor by 1mm, then 288 stages are needed. In the worst case for the topology shown in Figure 3, control elements located physically next to each other on silicon could occupy positions in the FIFO ring that are separated by 144 stages. Although FIFO rings do generate precise timing signals by locking the tokens into an evenly spaced pattern, a substantial amount of slack can still accumulate in the FIFO ring over 144 stages and render the timing signals useless.

2.3. Solution

Handshaking with four neighboring control elements instead of two allows the clock domain to be covered with a fewer number of elements. Instead of performing handshakes with the control elements to the left and right, the control circuitry handshakes with elements positioned along both the horizontal and vertical. Because the state on the conductors is retained for only a known, short period of time, we can dispense with the keepers.

Rather than using a serpentine route of 288 stages to clock a chip in a grid, a 12 × 12 grid of control built in the **Dynamic asP** style shown in Figures 4b, could be employed. Control elements are now separated by a maximum of 24 stages. Figure 9 shows an abstract 12 × 12 grid connected as proposed. The circles represent Pull-up elements, while the squares represent Pull-down elements, the vectors represent the state conductors. This figure is further explained later.

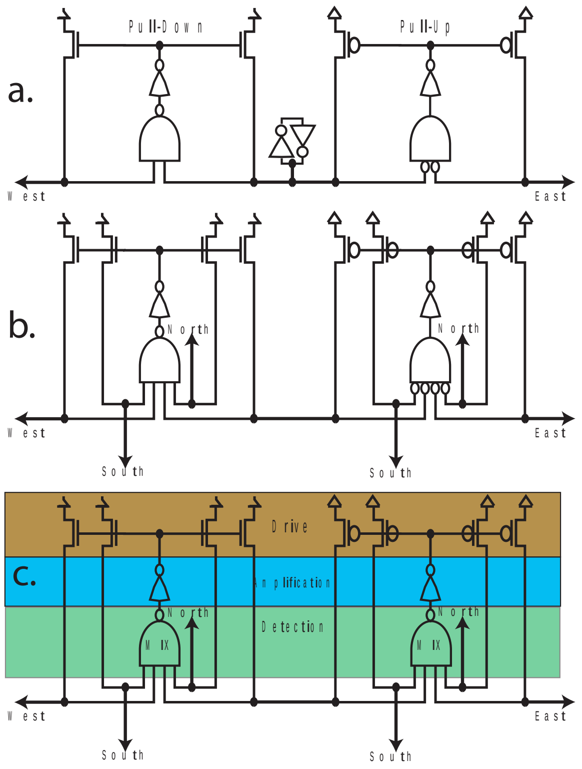


Figure 4.a-c Evolution from Dynamic asP to the Distributed Clock Generator.

This clock distribution apparatus works well as long as the signals on the state conductors are never too skewed. This grid is a two dimensional FIFO. Greatly skewed state conductors signify that the control tokens passing through it are greatly separated. The Charlie effect acts only when tokens are in close proximity to each other. The variable delay of the logic gate in the control element that gives rise to the Charlie effect does not exist over a large enough range of input separation times to yield the active skew correction desired.

A solution is to replace the logic gate with a phase mixing gate. A four input phase-mixer is shown in Figure 5. The phase mixing gate asserts when the cumulative average potential of the four inputs exceeds some threshold. Like the Analog C-element presented in [5], the delay of the gate as measured from the last arriving input increases as the separation time between the inputs decreases. We observe that control elements with phase mixers correct phase skews anywhere in the 360°range. Also, a phase mixer is easier to implement than NAND logic. This last step of evolution from **Dynamic asP** control to a structure used for clock distribution is shown in Figure 4c.

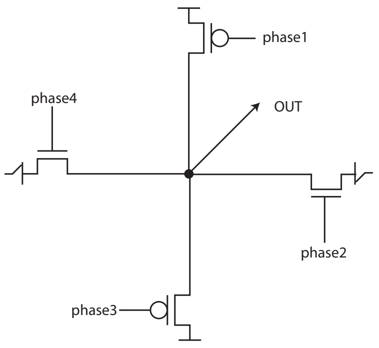


Figure 5. Four input phase mixer

Note that the inverters shown in Figure 4 signify a logical inversion. Any odd number of inversions can be used here depending on the target clock frequency. A ten gate delay clock period requires three levels of inversion in both control element types.

The clock distribution structure built from the control described is named the **Distributed Clock Generator**, or **DCG**. A 2×2 section from a **DCG** is shown in Figure 6. Notice the **DCG** uses two types of control elements. The Pull-up element synchronizes the four state conductors it senses on the rising edge. The Pull-down element synchronizes the four state conductors on the falling edge. Also notice that the **DCG** is not a number of scattered oscillators whose outputs are shorted. Rather the **DCG** is a single oscillator that is spread over the surface of the chip.

2.4. Geometry benefits

This control addresses the three primary concerns for a clock distribution strategy of power consumption, geometry, and design time that were identified at the beginning of this paper. Section 7.2 addresses the power requirement of the **DCG**. Section 4.1 presents a simple **DCG** design algorithm. Here we discuss the benefits of its geometry.

A number of benefits derive from the **DCG** geometry. The red and blue lines on Figure 6 represent power and ground rails from a grid power distribution system.

Routing The **DCG** is a distributed oscillator. Because the clock signal is not fanned out from a single point, the state conductors of the DCG are easily confined to preassigned routes. We suggest that the state conductors from the **DCG** be routed between the conductors that supply power and ground to the system.

EM shielding The supply conductors then provide free noise shielding.

Predictable current return paths The current return paths of the clock signal are then simple to trace and short because they run along the supply conductors. They are also consistent and predictable.

Split Inverter The drains of the PMOS and NMOS drive transistors in the **DCG** are separated by a long wire. The gates are driven by different signals, but ones that should be of the same phase. We call this the split inverter. In a normal inverter, the drains of the PMOS and NMOS transistors share the same node.

Notice that current moves in a single direction along the wire. When charging and discharging the load on a long wire, the split-inverter causes approximately equal currents on the wire. When a normal inverter charges and discharges the load on a long wire, it causes approximately equal but opposite currents on the wire. Equation 1 suggests that the split inverter induces a magnetic field with half the peak variations since its magnetic field curls in a single direction.

$$\vec{B} = \frac{\mu_0 \vec{I}}{2\pi r} \qquad (1)$$

3. Mechanics

Each control element can be divided into three components. These components are shown in Figure 4c.

Detection The detection component is simply the phase mixing gate. It uses two PMOS and two NMOS transistors. These transistors are labeled *pm1-pm4* for a phase mixing gate in a Pull-down element in Figure 7a. When used in the Pull-up element, a falling transition signals that the cumulative state of the four state conductors it senses are set LO and should be reset HI. The falling transition is a result of two actions. When the state conductors *sce* and *scw* in Figure 7a go HI, the NMOS transistors start discharging the output of the phase mixing gate. When the state conductors *scn* and *scs* go HI, the PMOS transistors stop conducting charge to the output of the phase mixing gate and cease to resist the falling transition. When the four state conductors oscillate in phase, the phase mixing gate is indistinguishable from two inverters connected in parallel. When the four state conductors are out of phase, the PMOS and NMOS transistors oppose each other. The delay of the mixing gate from an early arriving clock signal is long. The delay of the mixing gate from a late arriving clock signal is quick because the potential is removed from the supply voltage when the input arrives and less charge needs to be conducted to the output to cause a change in the logical value of the output.

When the four state conductors are out of phase, then the transistors in the logic gate burn static power. Fortunately, in the suggested topology, these logic gates are four gates of amplification removed from the large drive transistors, *nd1-nd4* and *pd1-pd4*, that charge the state conductors. Assume that the electrical amplification from gate to gate in the control elements for the **DCG** is chosen to be 4. The current at the output of the four input logic transistors that compose the phase mixing gate are then about three to the fourth power, or $1/256$, smaller than the currents of the drive transistors. This current is negligible.

Amplification The amplification component has an odd number of inverting amplifiers. It is detailed in Figure 7b. The amplification component serves three purposes.

1. The amplifiers take the relatively weak assertion signal provided by the detection circuitry and amplifies it to a strength that will drive four transistors capable of driving interconnect of lengths on the order of a millimeter.

2. The amplification component is responsible for resetting the clock network. The first amplifier in the amplification component is implemented with an asymmetric NAND gate and receives the $start$ signal. One input of the NAND gate connects to the output of the Detection component. The other input connects to the $start$ signal. Before the start signal asserts, the output of this NAND is HI. Pull-down elements are forced to drive all the state conductors LO while Pull-up elements disable their PMOS drivers.

3. The remaining amplifiers implement speed control functionality. A simple speed control places a series NMOS transistor in the pull-down path of one or more of the inverters in the amplification component. This transistor limits the amount of current available to discharge the gate's output node. Degrading the ability of these inverters to source current will slow the clock frequency. The speed control signal is a very low frequency analog signal requiring little care when distributing.

   Three is the most practical number of gates for amplification. One gate of amplification results in aggressive cycle times. A significant amount of skew begins to accumulate in five or more amplifiers is chosen.

Drive A single transistor drives each of the four state conductor. The gates of the four drive transistors occupy the same electrical node. This ensures that the four drive transistors act in unison and synchronize the signal on the four state conductors on the rising or falling transition. The drive transistors are left unencumbered from any speed control or logic. Their sole responsibility is efficiently driving the state conductors. The drive transistors are labeled *nd1-nd4* and *pd1-pd4* in Figure 7a.

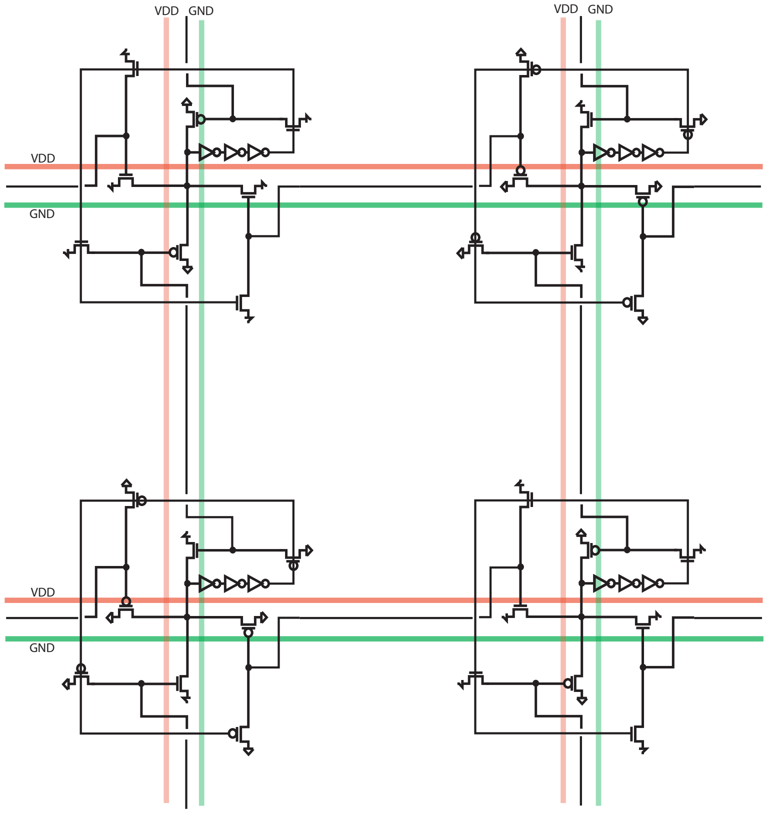


Figure 6. 2 x 2 section of a DCG grid

4. Design

4.1. Gate sizing

Below is a seven step algorithm to designing a **DCG**.

1. Choose a value for the electrical amplification from gate to gate in the control elements. Larger values result in lower power designs but result in signals with greater slew. The more slew of a signal the more vulnerable it is to noise.
2. Choose a value for the PMOS/NMOS transistor width ratio, γ.The best options are 1, 1.5 and 2. A γ of 1 results in Pull-up and Pull-down elements having the same footprint in the silicon. This choice is appealing from a geometric and regularity argument. Unfortunately, unless the chosen electrical amplification for the system is small, the rising transitions can be anemic and be vulnerable to noise.

   The minimal average delay for the rising and falling transition of an inverter is achieved with a γ value equivalent to the square root of the conductivity ratio of the electron and hole doped silicon used to construct the transistors [10]. This optimal value is about 1.6 in the process used for simulations in this paper. 1.5 is a good approximation of this value.

   A gamma of 2 balances the need for speed, tolerable slew rates, and circuits size.
3. Build a Spice model of the Pull-up and Pull-down control elements. Connect a wire between terminals N, S, E, W in one element to the similar terminal in the other control element. Be sure that wire uses an adequate RC or RCL model for the necessary wire length.

   Each gate's size should be paramaterized in Spice to be a function of the size of the PMOS drive transistor in the Pull-up control element. The last inverter in the amplification component of the Pull-up element should have a total transistor width equivalent to the transistor width of the four drive transistors divided by the chosen electrical amplification. The next gate should be reduced by another factor of the electrical amplification. The transistors in the phase mixer of the Pull-up element should have a total size equal to the size of the PMOS drive transistors divided by the electrical amplification raised to the fourth power. The transistors in the Pull-down elements should be scaled by from the transistors in the Pull-up element by $1/\gamma$.
4. This circuit should oscillate as is. Step the size of the PMOS transistor until the **DCG** oscillates at the target

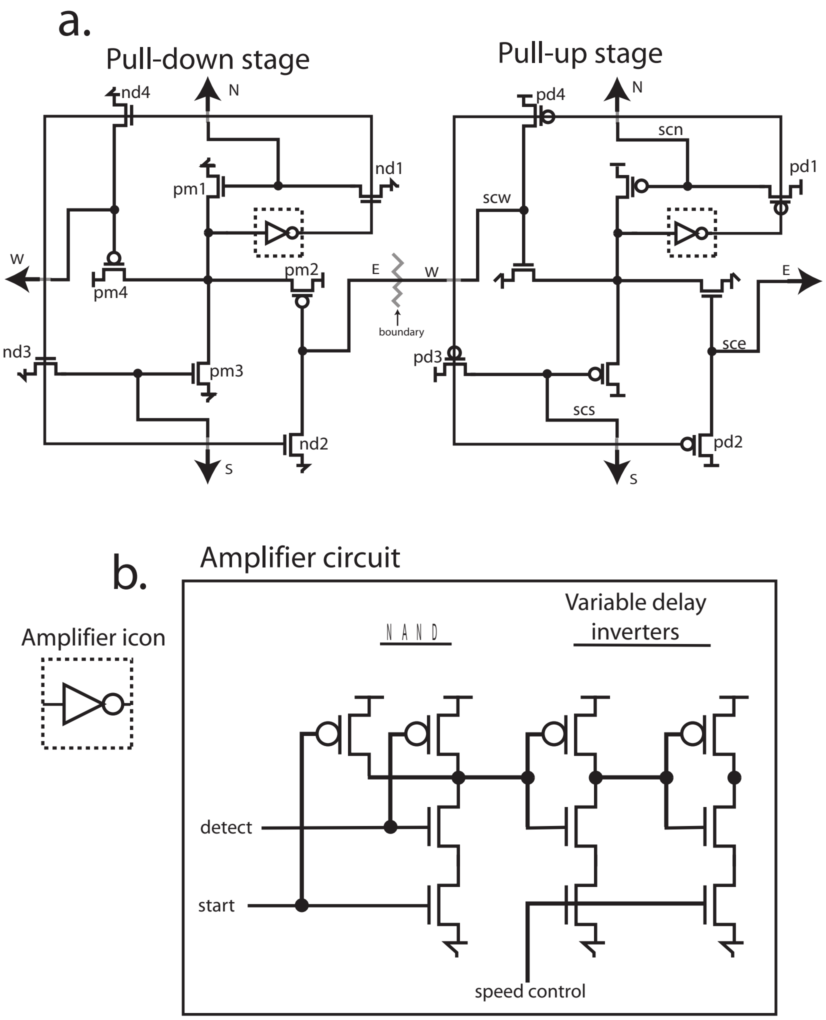


Figure 7.a-b Pull-up and Pull-down control elements in detail

cycle time. If the circuit doesn't reach the target cycle time, then a lower electrical amplification might give the speed desired. Another option is widening the state conductors wires.

5. Adjust the duty cycle by adjusting the speed control section of one type of control. If the duty cycle is too long, the delay through the Pull-up control elements needs to be increased by adjusting the speed control.
6. If the voltage swing at the clock output is insufficient, the electrical amplification chosen earlier needs to be reduced.
7. Take control element with resulting transistor sizes and arrange the control elements in a grid.

4.2. Interconnect optimization

The regularity of the **DCG** allows the interconnect to be optimized to a degree impractical in a clock tree. The dimensions of the state conductor that connects control elements need only be designed once and then these dimensions are duplicated many times throughout the design. Tapering the width of long interconnects along its length yields marginal but real speed and power savings [2]. Optimal tapering improves delay by up to 8% [1].

The optimal uniform state conductor width of length 1333μm when the PMOS driver was 42μm wide and the NMOS driver was 21μm is 1μm. We used a simple algorithm that segmented the state conductor into 9 sections. Each segment took a width centered around the optimal uniform width. Each segment was replaced by a simple RC model and the Elmore delay for the state conductor was calculated. The resulting optimal widths were 1.2, 1.1, 1.1, 1.2, 1.2, 1.2,1.2, 1.3,1.3. These are the widths of the state conductor starting from the side connected to the NMOS drive transistor. The replacement wire has an Elmore delay of 495 ps versus 525 ps for the wire of uniform width.

The state conductor can be drawn once and then duplicated throughout the system.

5. Mechanics

5.1. Initialization and starting

When the $start$ signal is LO or unasserted, a HI potential is applied to the gates of the transistors in the drive component of each control element. Consequently all state conductors are driven LO. The **DCG** begins oscillating when the $start$ signal asserts. It is not critical that the $start$ signal arrives at each control element simultaneously. If the $start$ signal is greatly skewed, then the state conductors initially oscillate out of phase but they quickly lock into phase soon after the $start$ signal distributes to all elements. The mechanism that forces the state conductors into the same phases is the Charlie effect which is enacted by the variable delay of the mixing gate in the detection component of the con-

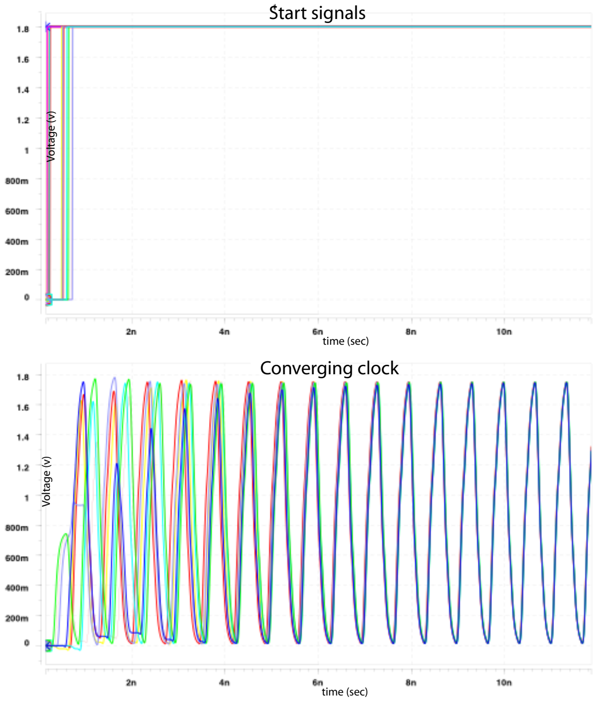


Figure 8. Top plot shows start signals asserting at random times, bottom plot shows six clock taps converging

Figure 8 shows two windows of a waveform viewer. The top window shows a number of start signals that are randomly chosen from a uniform distribution of a single clock period. These start signals are sent to the control elements in a 12×12 **DCG**. The bottom window shows the signals on a number of state conductors. Notice that the various clock signals are locked with respect to each other within 10ns.

5.2. Synchronization

When the combined potential on the four state conductors reaches a potential that causes the detection component to assert, the amplification component amplifies the asserted signal and causes the drive transistors to charge the state conductors to the opposite potential. Because the gates of the four drive transistors share the same electrical node, the four signals reset simultaneously irrespective of the phases of the signals when they arrived at the stage.

6. Hazards

6.1. Timing constraints

The drive from the state conductors is removed three gate delays after the detection circuitry signals that the potential on the state conductors has changed. The drive transistors first charge the state conductor wires near the detection circuitry. Then the charge spreads along the length of the wire. A small hazard exists if an insufficient amount of charge is sourced onto the state conductor to charge the whole length of the state conductor before the drive transistors are disabled. This is not a concern if the design methodology is disciplined and ensures that each gate in the **DCG** has the same electrical amplification required of it. Under this constraint the time allotted to charge the wire is always proportional to the capabilities of the drive transistors and this hazard is avoided.

6.2. Supply variations

Notice that current moves in one direction on the state conductors. A large amount of charge is sourced by the Pull-up elements and sunk by the Pull-down elements. If the power supply conductors are resistive then supply gradients exist between the two types of control. The supply voltage near the Pull-down elements is greater by an amount equivalent to the IR drop in the current return paths.

6.3. Mode lock

Mode lock is a stable system equilibrium in which the phase averaging mechanism used to couple the oscillators settles the oscillators in a non-zero phase relationship [8]. If two of the phases contributing to the average are of equal but opposite magnitude then an undesired and stable phase equilibrium occurs.

The criteria for avoiding mode lock in a two input phase mixer stipulates that the delay of the phase mixer must increase linearly for phase differences between +/-90° and decreases between 90°and 270°. This error function is easy to plot and visualize for a two input phase mixer but is substantially more difficult to formulate and visually represent with four inputs.

We present an empirical argument to show that the **DCG** is not susceptible to mode lock. Using HSpice, 100 simulations were ran of a 4x4 **DCG** whose clock period was 675ps. A test setup ensured that each element received its own $start$ signal. The individual $start$ signals asserted at a time picked from a uniform distribution between 0ps and 675ps. For this simulation transistor mismatch was not modeled. In all cases, the state conductors locked into phase by 20ns. The large number of initial phase relationships between the state conductors that did not excite the mode lock condition gives reasonable assurance that this behavior is not exhibited by the **DCG**.

7. Comparison

7.1. Implementation

A table describing electrical specifications for a typical clock distribution problem is found in the text book *Digital Systems Engineering* [3]. This paper reproduces the table as Table 1. This section describes a **Distributed Clock Generator** designed to provide a clock signal to a chip with these parameters. The target cycle time is 9 FO4 inverter delays or 675ps/cycle in the 180nm process used for simulation.

We chose to limit the electrical amplfication on each node in the clock network to 3.6. Values lower than this burn increasingly more power. Values greater than this become increasingly vulnerable to noise coupling. A disciplined clock distribution design methodology places a maximum electrical amplification of about five on all nodes of the circuit [9].

7.2. Performance

The relative performance of the **DCG** and a canonical H-tree are compared. We assume that the clock load is spread homogeneously over the surface of the microprocessor. The **DCG** has 264 clock taps. The H-tree has 256. The distance between control elements in the **DCG** is 1333μm. The final clock load is driven by two series inverters. Each inverter has an electrical amplification of 3.

Skew Figure 9 is an abstract representation of a 12 x 12 **DCG** grid constructed to clock the chip described by Table 1. The circles represent Pull-up elements, while the squares represent Pull-down elements. Each circle and square is labeled with two numbers that refer to its row and column. Vectors going from the Pull-up to the Pull-down elements represent state conductors. The arrows indicate the direction of current flow.

The skew measurement performed was relative to the vector coming from the element in row 5 and column 4. The skew is reported as the one sigma deviation skew. The numbers on the figure next to each vector are the average of the absolute value of the deviation of the rising edge and the absolute value of the deviation of the falling edge for the clock tap on that state conductor. The data was culled from a 30 run Monte Carlo simulation in HSpice. The skew is measured relative to the rising edge

Typical on-chip clock distribution problem	
Number of gates	10^6
Number of clock loads	5 x 10^4
Capacitance per clock load	20fF
Total clock capacitance	1nF
Chip dimensions	16 x 16 mm
Wire resistivity	0.07 $\frac{\Omega}{square}$
Wire area capacitance	0.13fF/μm^2
Wire length capacitance	0.08 fF/μm

Table 1. Reproduced from Dally and Poulton's *Digital Systems Engineering* [3]

Notice that the skew increases the greater the distance from the reference point. Also the increase is continuous, unlike clock trees where at some point two adjacent clock taps have maximally different paths from the root.

The simulations model a microprocessor built in a 180nm process. Only 60% of a chip this size can be reached in a single clock cycle with the clock period[6] simulated. Destinations beyond this distance need to be re-synchronized in a register. The skew seen at the top right corner are irrelevant because a signals couldn't reach those registers in a single clock cycle from the reference point.

The worst case local skew for an H-tree is where two adjacent clock taps route through maximally different routes. This route is through a string of eleven different inverters. Each branch has two inverters, the first inverter drives the second inverter. The second inverter drives into a branching point and drives two copies of the input to the next driver and the interconnect between. The first driver has an electrical amplification of 3 while the second driver has an electrical amplification of about 6. The one sigma deviation skew at the adjacent clock taps derived from maximally different routes is 27ps.

Power The **DCG** built for this task required 24,912μm of transistor width. A transistor presents about 1.9fF of capacitance for each micron of transistor. The approximate total of capacitance due to transistors for the system is 47.3pF. The **DCG** had 266 state conductors each 1333μm long and 1μm wide. The interconnect contributes 74pF of capacitance.

The H-tree required 37,520μm of transistor width. This represents 71.2pF of capacitance. The H-tree required 368,000μm of interconnect length as opposed

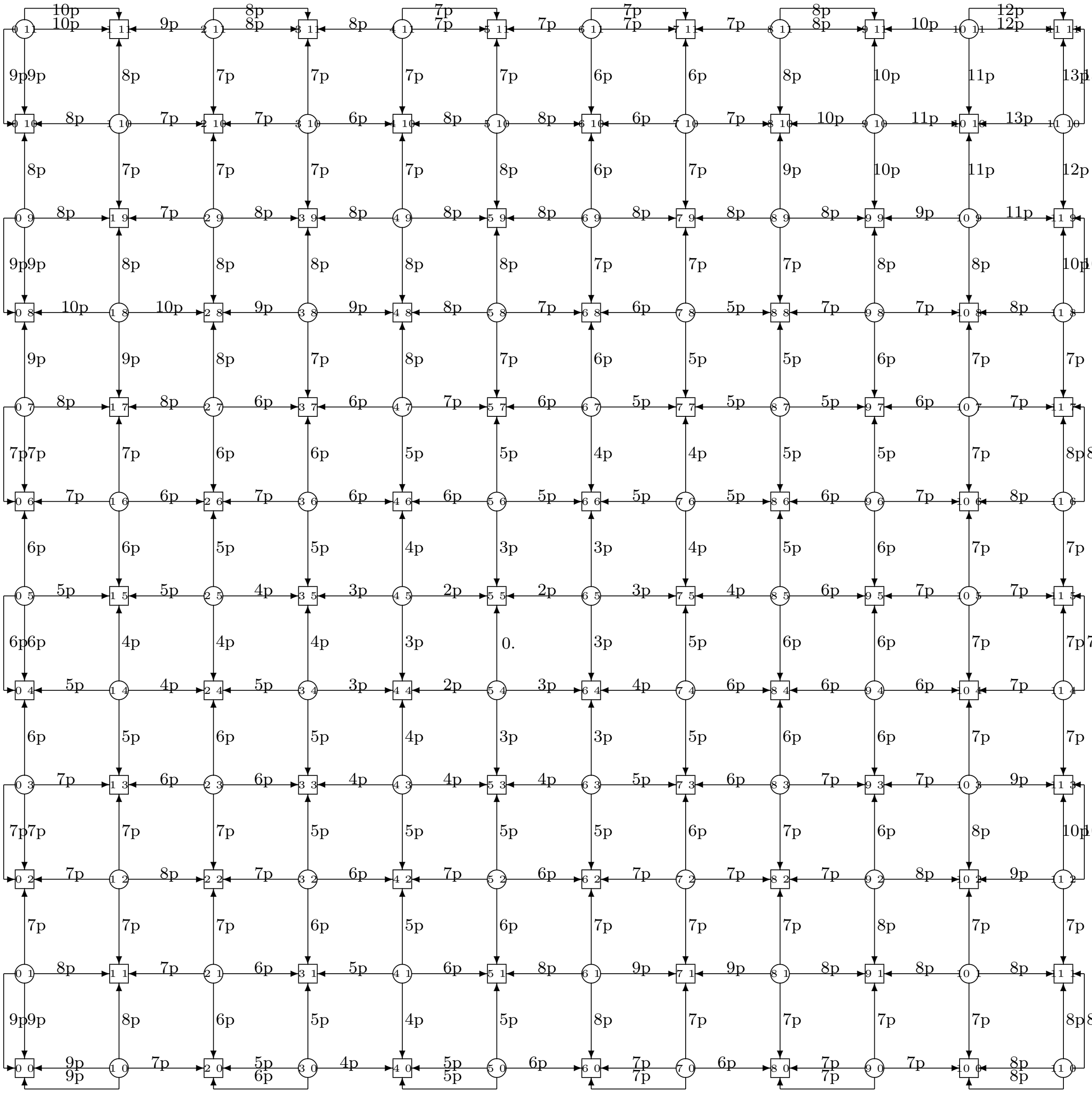

10p 8p 7p 7p 8p 12p
10p 9p 8p 8p 7p 7p 7p 7p 8p 10p 12p
9p9p 8p 7p 7p 7p 7p 6p 6p 8p 10p 11p 13p13p
8p 7p 7p 6p 8p 8p 6p 7p 10p 11p 13p
8p 7p 7p 7p 7p 8p 6p 7p 9p 10p 11p 12p
8p 7p 8p 8p 8p 8p 8p 8p 8p 9p 11p
9p9p 8p 8p 8p 8p 8p 7p 7p 7p 8p 8p 10p10p
10p 10p 9p 9p 8p 7p 6p 5p 7p 7p 8p
9p 9p 8p 7p 8p 7p 6p 5p 5p 6p 7p 7p
8p 8p 6p 6p 7p 6p 5p 5p 5p 6p 7p
7p7p 7p 6p 6p 5p 5p 4p 4p 5p 5p 7p 8p8p
7p 6p 7p 6p 6p 5p 5p 5p 6p 7p 8p
6p 6p 5p 5p 4p 3p 3p 4p 5p 6p 7p 7p
5p 5p 4p 3p 2p 2p 3p 4p 6p 7p 7p
6p6p 4p 4p 4p 3p 0. 3p 5p 6p 6p 7p 7p7p
5p 4p 5p 3p 2p 3p 4p 6p 6p 6p 7p
6p 5p 6p 5p 4p 3p 3p 5p 6p 6p 7p 7p
7p 6p 6p 4p 4p 4p 5p 6p 7p 7p 9p
7p7p 7p 7p 5p 5p 5p 5p 6p 7p 6p 8p 10p10p
7p 8p 7p 6p 7p 6p 7p 7p 7p 8p 9p
7p 7p 7p 6p 5p 6p 7p 7p 7p 8p 7p 7p
8p 7p 6p 5p 6p 8p 9p 9p 8p 8p 8p
9p9p 8p 6p 5p 4p 5p 8p 7p 7p 7p 7p 8p8p
9p 7p 5p 4p 5p 6p 7p 6p 7p 7p 8p
9p 6p 5p 7p 7p 8p

Figure 9. Skew expectation over the surface of microprocessor due to transistor mismatch

to 353,000μm of length for the **DCG**. All levels except for the final 'H' used 3μm wide wire. The total wire capacitance for the H-tree is 104.8pF.

The total H-tree capacitance is 176pF while the total **DCG** capacitance is 121pF. If both systems used the same supply voltage and identical clock frequencies, then the **DCG** would consume $\frac{121}{176} = 0.68$ of the power as the H-tree.

Speed control The simple speed control circuitry yields a wide range of clock periods. The relationship between speed control voltage and the resulting period is shown in Figure 10

Jitter Each transistor connected to the supply voltage in the **DCG** was connected to a unique dirty supply. The dirty supplies were created by connecting the dirty node to two voltage sources, each through a very large transparent transistor.The first supply swung between 0.9×V_{supply} and 1×V_{supply} with a frequency of 9×FO4 delays. The second supply had the same voltage swing but the period was chosen from a uniform random distribution between 150ps and 250ps. The first supply simulates noise contributed from the output resistance of the supply while the second supply simulates higher frequency noise sources. HSpice performed a thirty run Monte Carlo simulation and the 1 sigma expected jitter offset between clock taps found in the same row but in adjacent columns was 1.3ps.

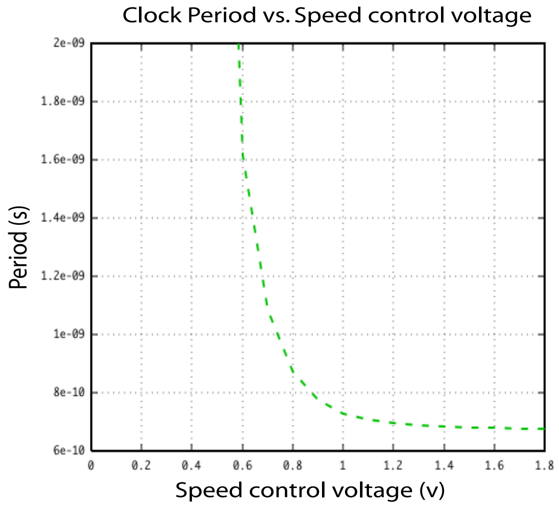


Figure 10. DCG period as a function of speed control voltage

For comparison the same test was run using a string of amplifiers with an electrical gain of four between gates in a control element. Interestingly, the dirty voltage supply meant that the signal was filtered away before the eleventh amplifier. Eleven is the number of levels of amplification from root to leaf in the comparison between the **DCG** and the H-tree above. When the clock was slowed from 9 to 11 FO4 periods the signal correctly amplified through the inverter string. The 1 sigma expected jitter offset here was 0.1ps. This is significantly better than the **DCG** but it should be noted that the noise modeled should have been greater in the H-tree because it makes greater demands on the power supply. Yet the simulations modeled the same amount of noise in both systems.

8. Conclusion

Dynamic asP provides the foundation for an efficient, low skew, and low power clock distribution solution. The **DCG** retains the drive efficiency and spartan use of hardware of **Dynamic asP**. The inputs are phase mixed instead of ANDed, allowing for a much greater range of skews that can be actively corrected. The control element handshakes with four other elements instead of two. This change yields a simple grid topology.

The grid topology prevents the confounding of clock signal routes with other signal routes, obviating the need for clock-tree rebalancing, and facilitating timing closure. Clock tree topologies necessarily have adjacent leaves where the clock signal is copied through maximally different paths, the **DCG** is free of this constraint. The **DCG** does not exhibit mode lock. The **DCG** obeys good design principles, design once and duplicate. This allows for the relatively simple duplicated part to be highly optimized, even the state conductor connecting control elements can be tapered along its length for marginal, yet real performance gains. The current paths for the **DCG** are easily traced, resulting in predictable inductances. The conductors of the clock signal are routed near the power distribution rails resulting in short return paths and free shielding.

References

[1] C.-P. Chen. Performance-driven interconnect optimization (dissertation). In *IEEE/ACM Design Automation Conference*, volume 34, 1997.

[2] C.-P. Chen, H. Zhou, and D. F. Wong. Optimal non-uniform wire-sizing under the elmore delay model. In *ICCAD*, pages 38–43, 1996.

[3] W. Dally and J. W. Poulton. *Digital Systems Engineering*. Cambridge University Press, 1998.

[4] J. C. Ebergen, S. Fairbanks, and I. E. Sutherland. Predicting performance of micropipelines using Charlie diagrams. In *Proc. Intern. Symp. on Advanced Research in Async. Circuits and Systems*, pages 238–246, 1998.

[5] S. Fairbanks and S. Moore. Analog micropipeline rings for high precision timing. In *Proc. Intern. Symp. on Advanced Research in Async. Circuits and Systems*, pages 41–50. IEEE Computer Society Press, Apr. 2004.

[6] D. Matzke. Will physical scalability sabotage performance gains. In *IEEE Computer*, volume 30, 1997.

[7] C. Molnar and S. Fairbanks. Control structure for a high-speed asynchronous pipeline. *US patent*, August 10, 1999. #5,937,177.

[8] G. Pratt and J. Nguyen. Distributed synchronous clocking. In *IEEE Transactions on Parallel and Distributed Systems*, March 1995.

[9] Restle, P.J. et al. A clock distribution network for microprocessors. *IEEE Journal of Solid State Circuits*, pages 792–799, May 2001.

[10] I. Sutherland, B. Sproull, and D. Harris. *Logical Effort: Designing Fast CMOS Circuits*. Morgan Kaufmann Publishers, Inc., 1999.

[11] I. E. Sutherland. Micropipelines. *Communications of the ACM*, 32(6):720–738, June 1989.

[12] A. J. Winstanley, A. Garivier, and M. R. Greenstreet. An event spacing experiment. In *Proc. Intern. Symp. on Advanced Research in Async. Circuits and Systems*, pages 47–56, Apr. 2002.

[13] P. Zarkesh-Ha. Global interconnect modeling for a gigascale system-on-a-chip (GSoC). Technical report, Georgia Institute of Technology, Feb. 2001.