Report Number: WUCS-88-36

1988-12-01

Design of a Clock Generator Chip

Tony Y. Mazraani

This report describes the design of a Clock Generator Chip. The purpose of this chip is to generate a non-overlapping three-phrase clock from single 50% duty-cycle clock. The design includes combinational logic, VLSI layout, and logic and timing simulations.

DESIGN OF A CLOCK GENERATOR CHIP

Tony Y. Mazraani

WUCS-88-36

December 1988

Department of Computer Science
Washington University
Campus Box 1045
One Brookings Drive
Saint Louis, MO 63130-4899

Abstract

This report describes the design of a *Clock Generator Chip*. The purpose of this chip is to generate a non-overlaping three-phrase clock from a single 50% duty-cycle clock. The design includes combinational logic, VLSI layout, and logic and timing simulations.

This work supported by Bell Communications Research, Bell Northern Research, Italtel SIT, NEC and National Science Foundation grant DCI-8600947.

Contents

List of Figures

DESIGN OF A CLOCK GENERATOR CHIP

Tony Y. Mazraani
tonym@wuccrc.wustl.edu

1. Introduction

This report describes the design of a *Clock Generator Chip* (CLKGEN) which is intended to generate a non-overlaping three-phase clock from a single 50% duty-cycle main clock. Throughout this report, I am going to refer to the generated clock phases as Φ_1, Φ_2, and Φ_3 and to the main clock as *System Clock* (SYSCLK). The design of CLKGEN involves combinational logic, VLSI layout, and logic and timing simulations.

2. Design Motivation

All VLSI chips that go into building a prototype for the BPN [Tu86] switch are designed to be driven by a non-overlaping three-phase clock. This clock is going to be generated externally and distributed appropriately to all boards. Using the design of CLKGEN, we are investigating the possibility of generating the required three clock phases using only one main clock, SYSCLK. If this design proves to be successful, we will consider the advantages and disadvantages of making CLKGEN be part of every chip. That is, all chips will be driven by a single clock, SYSCLK, instead of three.

3. Clock Generator Design

The objective of this design, therefore, is to generate Φ_1, Φ_2, and Φ_3 from one main clock, SYSCLK. Figure 1 is an illustration of how the generated clock phases ideally look.

This design can be implemented using a variety of approaches. The approach employed in this chip was suggested by Pierre Costa (see Figure 2). A 50% duty cycle clock, SYSCLK, and an appropriate combination of delays and signal inversion are used to generate the required clock phases.

As is shown in Figure 2, Φ_3 is a delayed version of SYSCLK. We use the rising edge of Φ_3 to initiate a Φ_1 pulse and the rising edge of $\Phi_{3delayed}$ to terminate this pulse. Then, we use the rising edge of $\overline{\Phi}_{1delayed}$ to initiate a Φ_2 pulse. This pulse is then terminated by the rising edge of $\Phi_{2delayed}$.

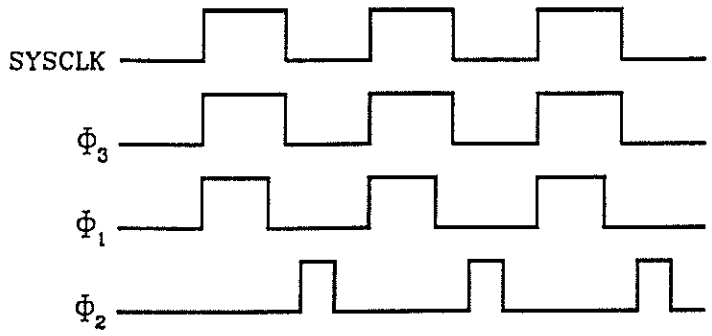


Figure 1: Clocks Generated from SYSCLK

4. Design Specifications

The BPN switch employs two clock frequencies, approximately 25MHz and 50MHz. So, the design of CLKGEN is particularly tailored to meet both frequency specifications. A small deviation from both frequencies, if desired, is also possible by means of a fine frequency tuning that is embedded in CLKGEN.

The timing specifications for Φ_1, Φ_2, and Φ_3 are as follows (see Figure 3). For an operating frequency of 50MHz, the pulse widths of Φ_1, Φ_2, and Φ_3 are 4 *nsec*, 10 *nsec*, and 12 *nsec* respectively. The widths of Φ_{12} (the distance between the falling edge of Φ_1 and the rising edge of Φ_2) and Φ_{21} (the distance between the falling edge of Φ_2 and the rising edge of Φ_1) are 6 *nsec* and 4 *nsec*

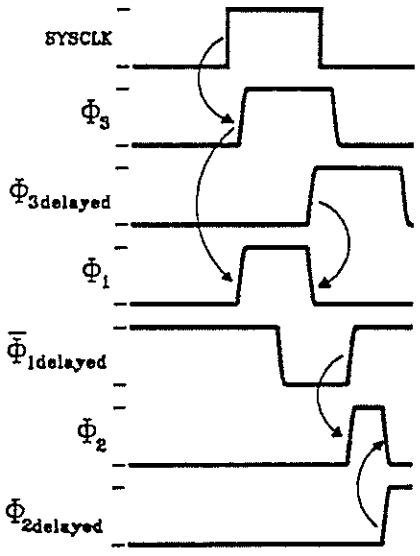


Figure 2: Design Approach of CLKGEN

respectively. For an operating frequency of 25MHz, the widths mentioned previously are 6.67 *nsec*, 16.6 *nsec*, 20 *nsec*, 10 *nsec*, and 6.67 *nsec* respectively.

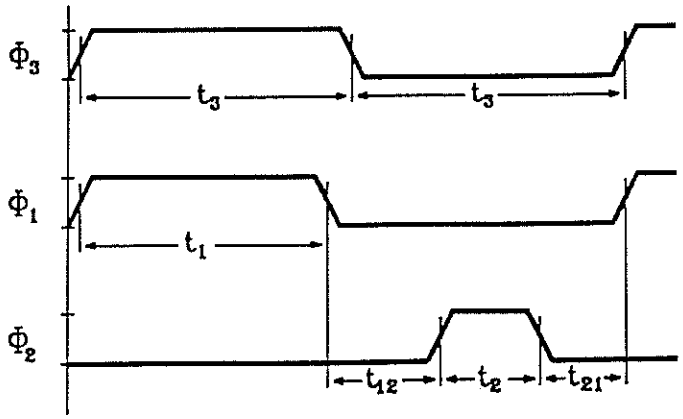


Figure 3: Clock Timing Specifications

5. Clock Generator Chip

A block diagram of CLKGEN is shown in Figure 4. This diagram follows the design approach illustrated in Figure 2. A description of the signals used in this circuit is as follows:

DA0, DA1	decoder input signals. They control the width of Φ_1.
DB0, DB1	decoder input signals. They control the width of Φ_{12}.
DC0, DC1	decoder input signals. They control the width of Φ_2.
LH	Low-High frequency signal (LH=0, low frequency; LH=1, high frequency).
IX	Internal/External signal. It controls the operation of the bidirectional pads (IX=0, output pad; IX=1, input pad).
SYSCLK	SYStem CLocK, main clock used to generate the three-phase clock.
PHI1I/0, PHI2I/0	Φ_1 and Φ_2 which are used as input and output signals. When used as output signals, they are generated by CLKGEN (same as PHI1 and PHI2). When used as input signals, they drive PHI1 and PHI2 externally. In this case, CLKGEN is totally isolated from internal circuitry.
PHI1TST, PHI2TST	used for computing delays through input and output pads. When IX=1, input signals into PHI1I/O and/or PHI2I/O follow a cut-through path directly to PHI1TST and PHI2TST respectively. In this case, the communication path consists of an input pad, a simple 2-input multiplexer, and an output pad.

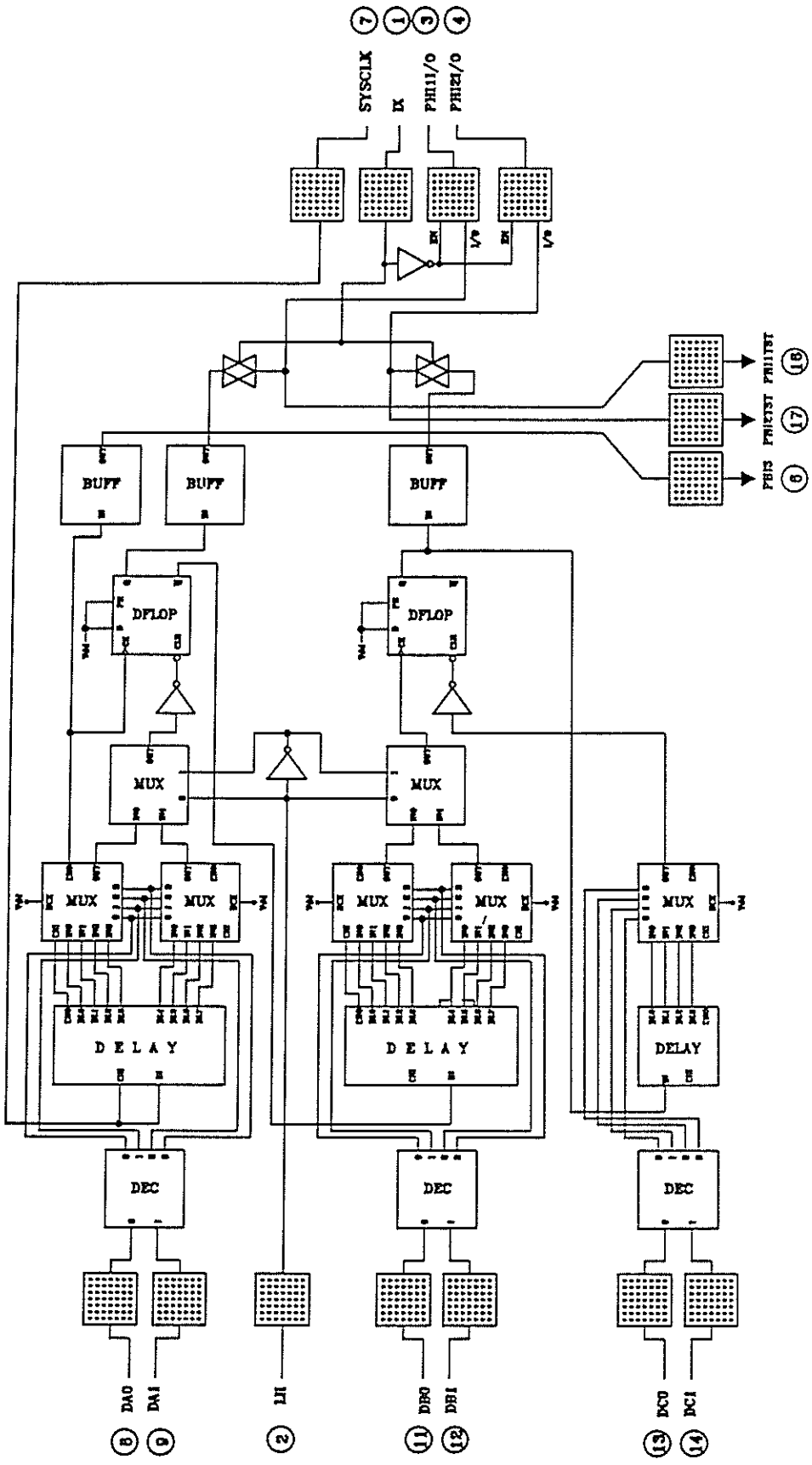


Figure 4: Block Diagram of CLKGEN

6. Circuit Description

This section describes the implementation of all blocks in CLKGEN circuit as well as different timing simulation results that were obtained using SPICE [ScMa86] and FACTS [MCNC].

6.1. 2-Input Decoder

A gate implementation of the 2-input decoder is shown in Figure 5.

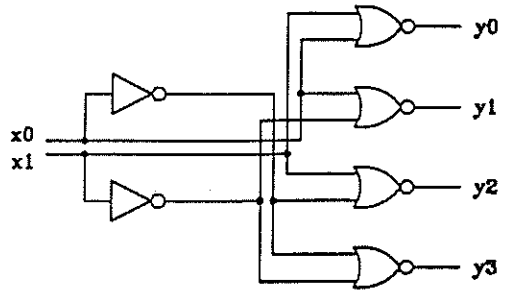


Figure 5: Implementation of a 2-Input Decoder

This decoder is used to drive the multiplexer which in turn selects the appropriate delay that meets the specifications.

6.2. 5-Input Multiplexer

A transmission gate implementation of the 5-input multiplexer is shown in Figure 6.

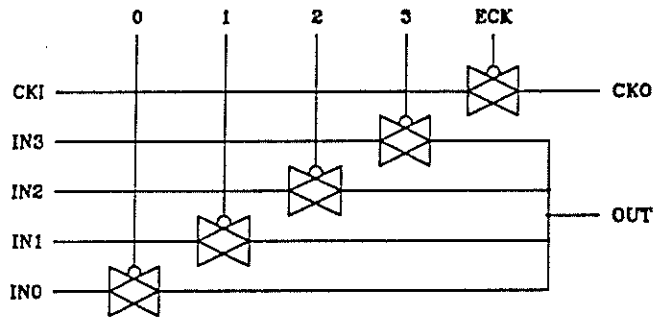


Figure 6: Implementation of a 5-Input Multiplexer

This multiplexer operates on inputs IN0, IN1, IN2, and IN3 in a standard way; it multiplexes them into one output, OUT. The input CKI runs across the multiplexer and is enabled by ECK.

6.3. 2-Input Multiplexer

A transmission gate implementation of the 2-input multiplexer is shown in Figure 7.

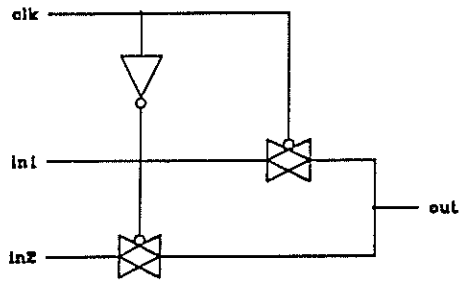


Figure 7: Implementation of a 2-Input Multiplexer

6.4. Delay Flip Flop

The implementation of the delay flip flop is adapted from [LLC85] and is shown in Figure 8.

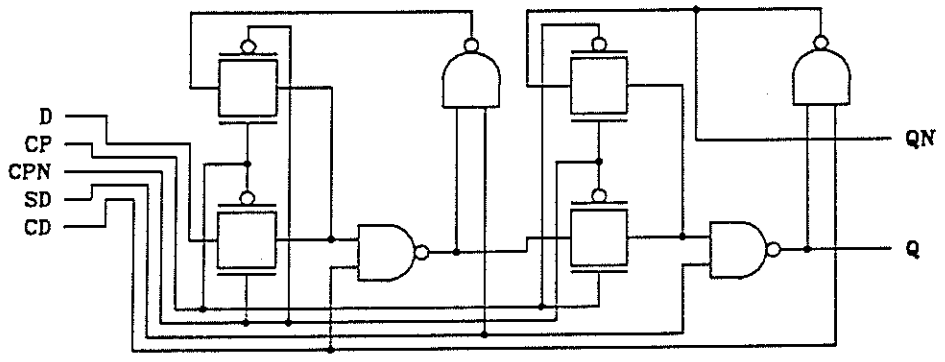
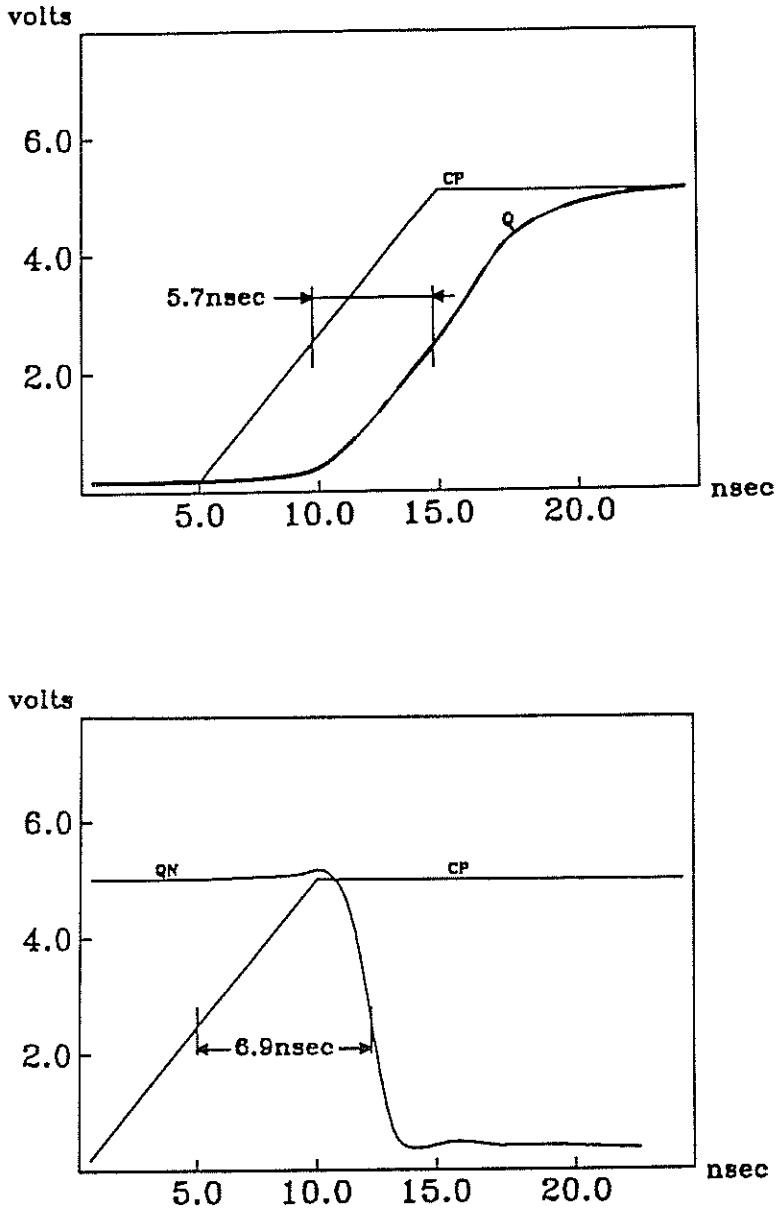


Figure 8: Implementation of a Delay Flip Flop

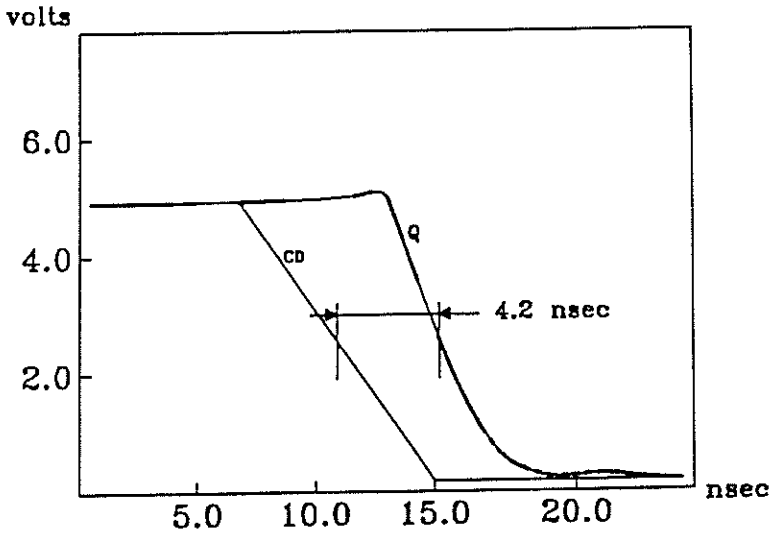
This circuit employs a Clear Direct (CD), a Set Direct (SD), and two unbuffered clocks (CP and CPN). The delay through the flip flop as a function of input clock and clear signal is depicted in Figure 9 (a) and (b). The maximum delay, 6.9 *nsec*, is from input clock CP to inverted output QN. The flip flop can operate at maximum frequencies up in the neighborhood of 50 MHz. Note that

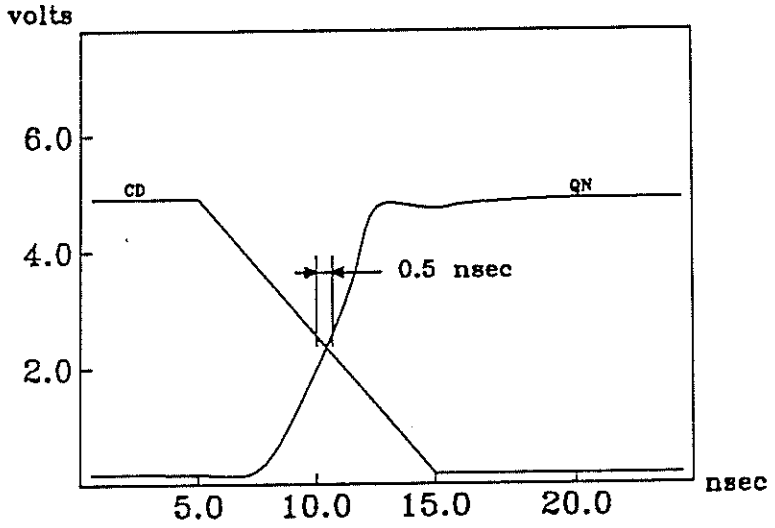
this result is derived from simulations and is very optimistic. The real speed of the flip flop will chiefly depend on the chip's fabrication process.



(a)

Figure 9: Delay Through Flip Flop





(b)

Figure 9: Delay Through Flip Flop

6.5. Delay Circuit

The delay circuit consist of a simple chain of inverters as shown in Figure 10. Each delay unit consists of two inverters.

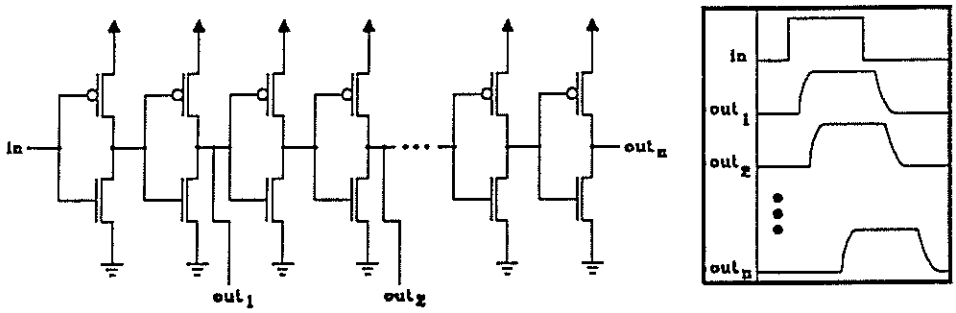


Figure 10: Delay Circuit Implementation

The delays used in CLKGEN were determined by a series of simulations using FACTS. Several delays are used instead of one to provide a fine delay tuning which is helpful when a fine frequency change is desired. Also, this range of delays helps compensate for any discrepancies between simulation results and real time testing results.

6.6. Buffer Circuit

The buffer circuit consists of two inverters in series with exponentially sized transistors. This is depicted in Figure 11.

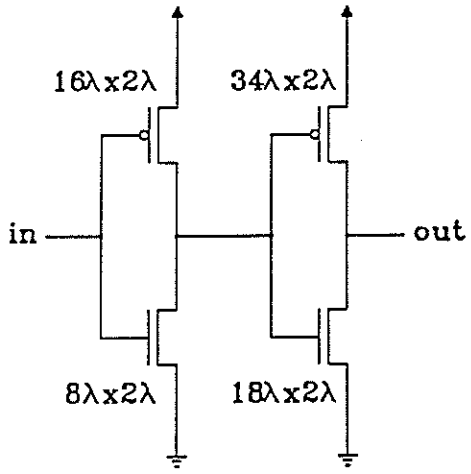


Figure 11: Buffer Circuit

The delay through the buffer circuit is shown in Figure 12. This delay is constant for both possible input signal levels and is equal to 0.68 *nsec*.

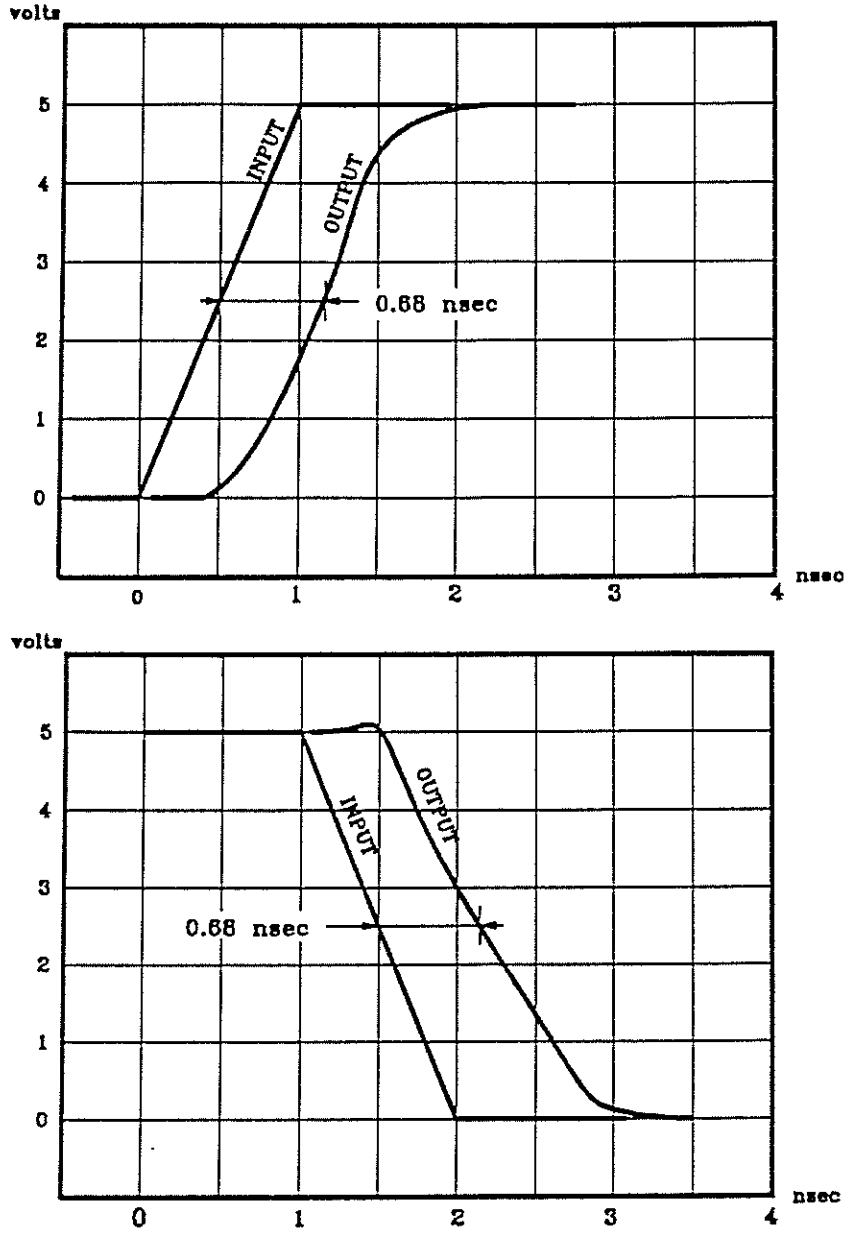


Figure 12: Delay through Buffer Circuit

7. Clock Generator Tiny Chip

A MAGIC file showing the cell level of CLKGEN is depicted in Figure 13. The VLSI layout of CLKGEN is shown in the upper half portion of Figure 14. It is contained in a tiny chip, having a die size of $1.1mm \times 2.3mm$. This chip is being shared with another project. The transistor count for CLKGEN is 546 transistors.

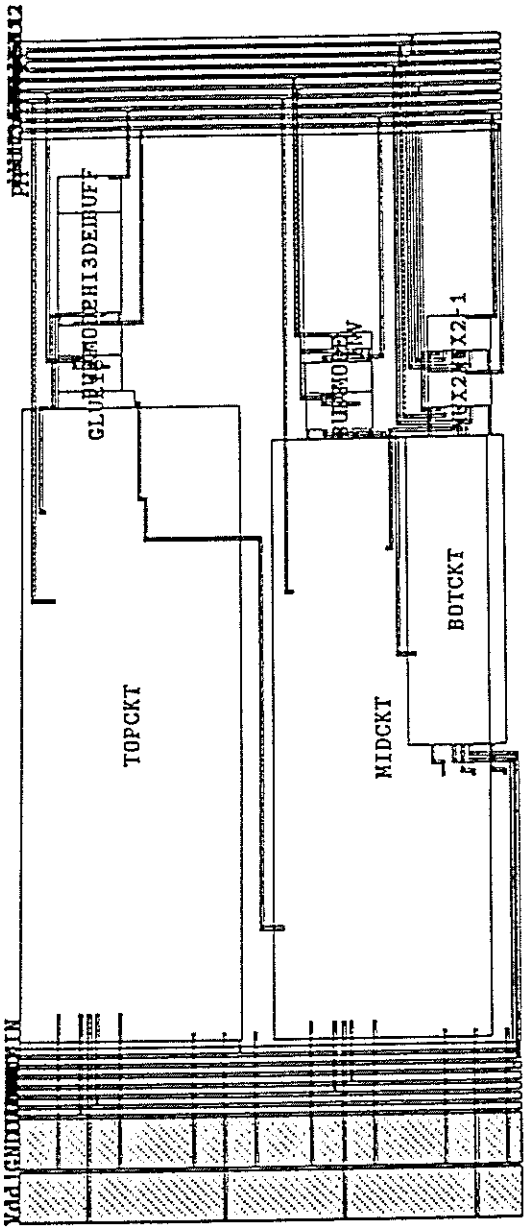


Figure 13: Cell level Diagram of CLKGEN

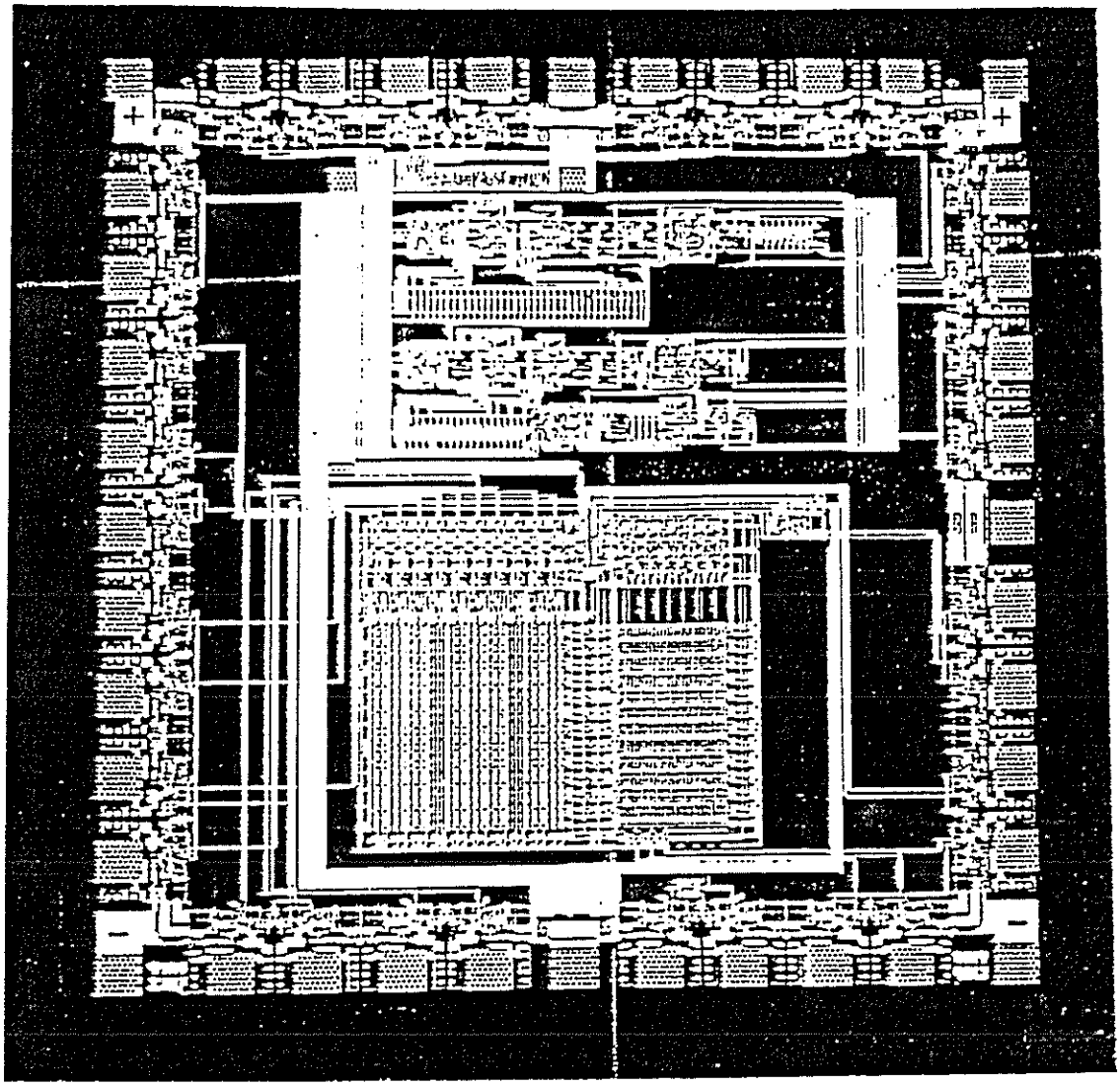


Figure 14: Tiny Chip Containing CLKGEN

8. Circuit Design Verification

All simulations were performed on low level components as well as the whole chip. Circuit-level simulation was performed using FACTS and SPICE. Logic-level simulation was performed using VESIM, a front-end interface to ESIM [ScMa86]. A simulation run using FACTS on the chip level is shown in Figure 15. It depicts the three generated clock phases at a frequency of approximately 25 MHz.

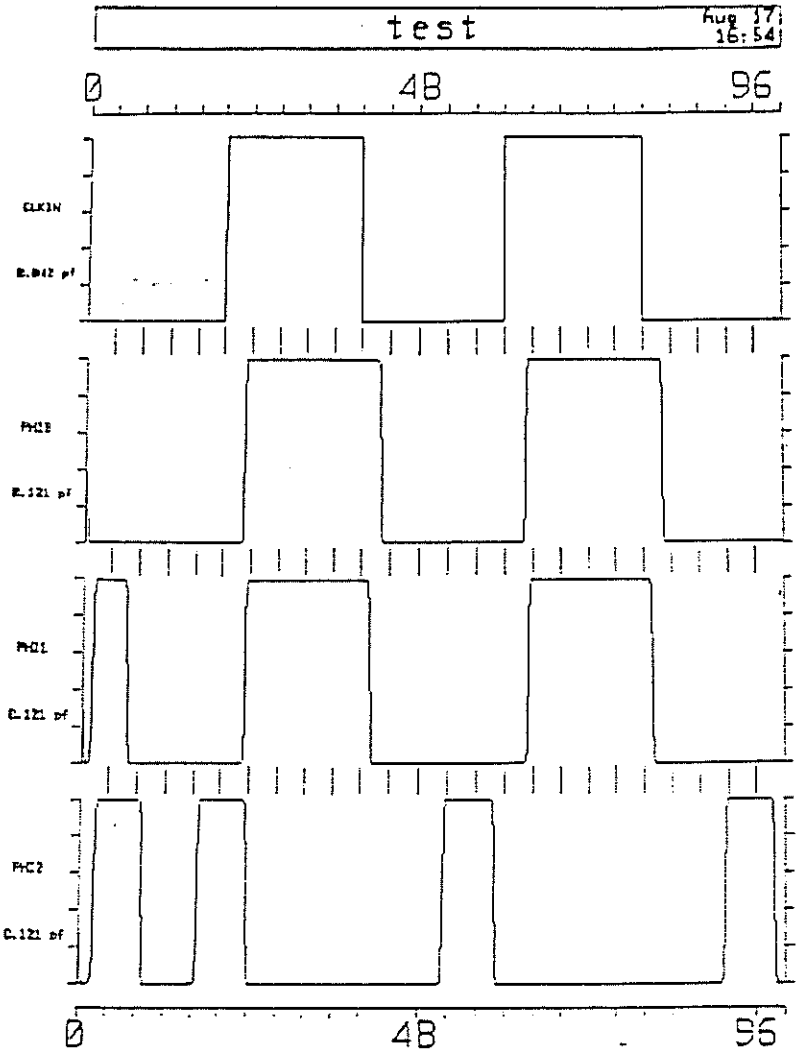


Figure 15: Example Chip Simulation Using FACTS

9. Chip Testing

This chip was fabricated by MOSIS and fit in a Dual-In-line Package. The fabrication process was done in mid October, 1988. We tested the chip using the following testing tools: *E-H Research Laboratories* pulse generator with a maximum frequency of 100 MHz; *Tektronix DAS 9100* Digital Analysis System; *Tektronix 2465A CT* oscilloscope with a maximum bandwidth of 250 MHz; and *Tektronix 91P32* pattern generator module.

The main clock was generated by the pulse generator. DAS was used to generate signals IX and LH. All other control signals, which are either 0 or 5 volts, were generated using dip switches which connect those signals as necessary to either ground or power line.

The statistics collected from the chip testing are illustrated in Figure 16. W_i is the pulse width of clock Φ_i. $\overline{W_3}$ is equal to period of Φ_3 minus W_3. For an ideal 50% duty cycle clock, W_3 is equal

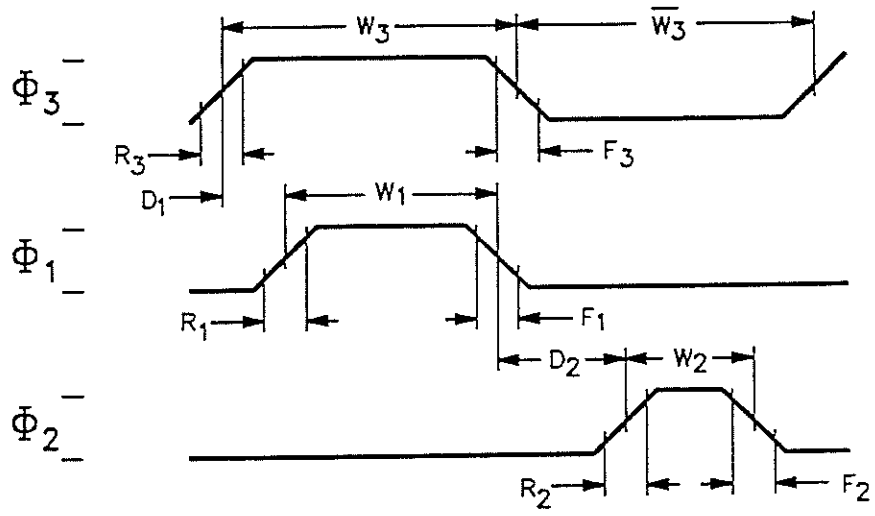


Figure 16: Statistics Collected from Chip Testing

to $\overline{W_3}$. R_i and F_i are respectively the rise time and fall time of clock Φ_i. D_1 is the delay between the rising edge of Φ_3 and the rising edge of Φ_1. D_2 is the distance between the falling edge of Φ_1 and the rising edge of Φ_2. The numbers W_i and D_i are computed at the 50% amplitude point; R_i and F_i are computed from 10% to 90% amplitude points.

Statistics collected for three different operating frequencies are shown in Figure 17. The results are acceptable (i.e. close enough to the specifications) for frequencies up to 20 MHz. For frequencies above 20 MHz, the chip behaves poorly. For example, at a frequency of 26 MHz, $W_1 = 2W_3$ which is highly different from what the chip specifications require, that is $W_1 \simeq 0.83W_3$. At this point, we find that that the D-flop is a delay bottleneck for frequencies above 20 MHz. Typical generated clocks at a frequency of 12.5 MHz are shown in Figure 18.

FREQ (MHz)	Pulse Width ($nsec$)			
	W_1	W_2	W_3	W_3
12.50	22.75	11.22	42.10	37.90
20.00	22.00	10.80	20.00	29.00
26.00	25.00	12.00	12.50	25.00

FREQ (MHz)	Rise Time ($nsec$)			Fall Time ($nsec$)			Delay ($nsec$)	
	R_1	R_2	R_3	F_1	F_2	F_3	D_1	D_2
12.50	8.40	6.86	7.40	6.40	5.00	5.60	6.94	11.40
20.00	8.15	6.30	7.40	6.90	5.75	5.00	8.22	11.00
26.00	8.00	6.80	7.10	6.80	5.50	5.70	4.90	9.00

Figure 17: Statistics for Three Different Operating Frequencies

The parameter D_1 is supposed to be in the neighborhood of zero which obviously is not the case for D_1 values shown in Figure 17. During the design phase of the chip, we added some delay circuitry to take care of synchronizing the rising edges of Φ_1 and Φ_3, thus bringing the value of D_1 close to zero. This delay was computed using SPICE and FACTS. What should have been done instead is to provide a range of delays spread around the already computed delay.

In order to test different delays used in this chip, we collected statistics about W_1, W_2, and D_2 for all possible decoder outputs at a frequency of 12.5 MHz (see Figure 19). At this point, we noticed an error in decoder DA: W_1, as shown in Figure 19, does not monotonically increase with increasing decoder output. We checked the MAGIC layout of CLKGEN and found out that the error was in routing the signal DA1 to the input pad. DA1 was routed to the inverted input (INB) of the pad instead of IN.

One result worthy of note is the delay through two inverters in series and how it compares with simulation results. In the case of D_2, one delay step consists of two inverters in series with transistor sizes of $8\lambda \times 2\lambda$. As shown in Figure 19, this delay is equal to 1.35 *nsec* which is equal to the delay obtained by FACTS when measured at the 90% amplitude point.

Summary of Testing.

- Results are adequate for frequencies up to 20 MHz.
- Delays obtained from testing are equal to delays obtained from FACTS when measured at 90% amplitude point $\pm$ 5% tolerance.
- D-flop becomes a delay bottleneck for frequencies above 20 MHz. Therefore, alternative designs for speeding up the D-flop should be considered.
- Delay circuitry should be added to Φ_3 in order to synchronize its rising edge with the rising edge of Φ_1.
- An error is found in decoder DA. The input signal DA1 was incorrectly routed to the inverted input of the pad.
- At some frequencies, the position of Φ_2 with respect to Φ_1 (i.e. D_2 in Figure 19) is slightly off timing specifications. This can be solved by providing a wider range of values for D_2.

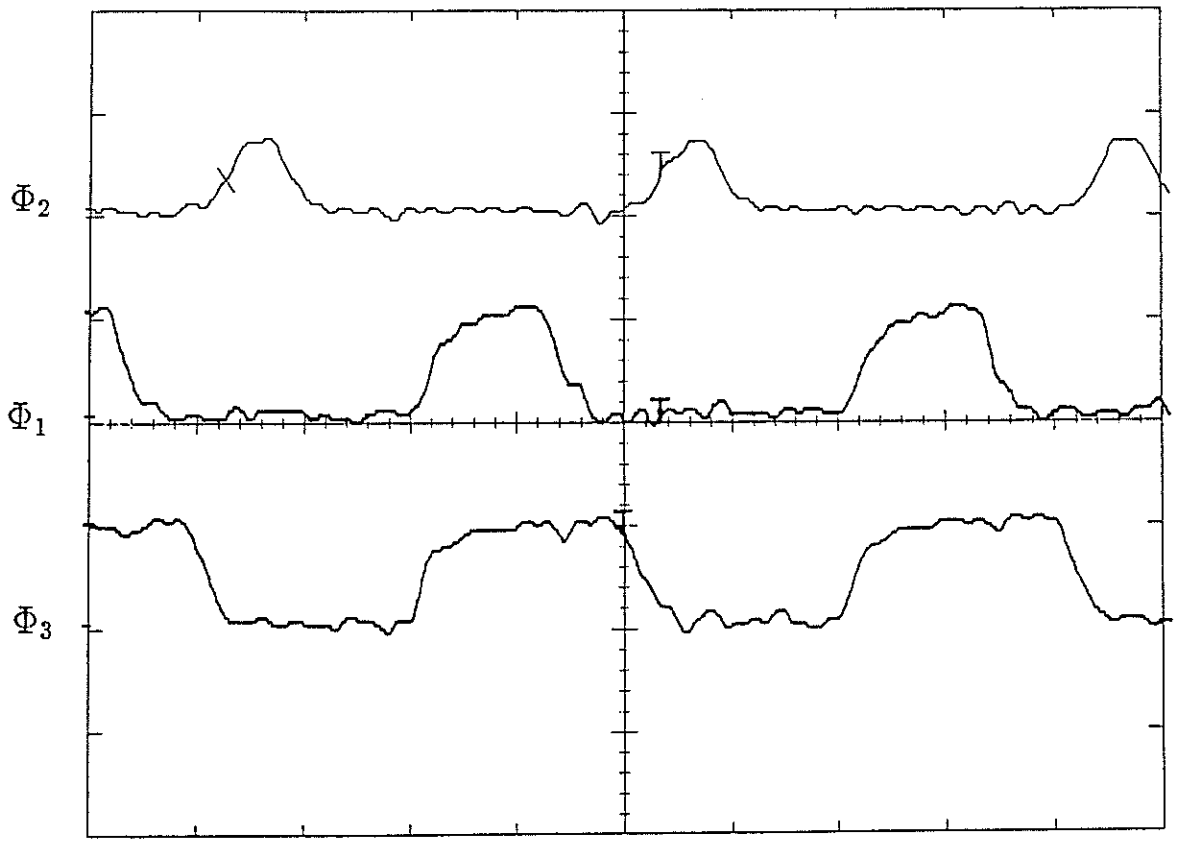


Figure 18: Observed Clocks at 12.5 MHz

DA1	DA0	W_1	DB1	DB0	W_2	DC1	DC0	D_2
0	0	20.34	0	0	4.75	0	0	11.65
0	1	22.75	0	1	6.00	0	1	13.00
1	0	15.38	1	0	10.22	1	0	14.35
1	1	17.86	1	1	11.22	1	1	15.70

Figure 19: All Possible Values for W_1, W_2, and D_2 at 12.5 MHz

10. Conclusion

The design of CLKGEN is intended to help us investigate the possibility of generating a non-overlaping three-phase clock from a single 50% duty-cycle main clock. Testing CLKGEN provided us with means to compare simulation and testing results as well as developing a clear idea about how to use simulation tools more efficiently in future projects.

References

[LLC85] *CMOS Macrocell Manual*, LSI Logic Corporation, Milpitas, CA, 1985.

[MCNC] *VIVID System FACTS Fast Circuit Simulator Designer Tutorial Version 1.3*, and, *VIVID System FACTS Fast Circuit Simulator Designer Reference, Version 1.3*, Microelectronics Center of North Carolina, Research Triangle Park, N.C., 1985.

[ScMa86] Scott, Walter S., Robert N. Mayo, Gordon Hamachi, and John K. Ousterhout, ed., *1986 VLSI Tools*, University of California at Berkeley, Computer Science Division (EECS), Report No. UCB/CSD 86/272, Berkeley, Calif., Dec. 1986

[Tu86] Turner, Jonathan S., "Design of a Broadcast Packet Network," *Proceedings of IEEE Infocom'86*, IEEE Computer Society, Los Angeles, April 1986, pp. 667-675.