12-31-1990

Digital correlators for spread spectrum communications

Raymond B. Chadwick
New Jersey Institute of Technology

Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be "used for any purpose other than private study, scholarship, or research." If a, user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of "fair use" that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select "Pages from: first page # to: last page #" on the print dialog screen

ABSTRACT
Digital correlators for
Spread Spectrum Communication

by
Raymond B. Chadwick

The design and application, in spread spectrum communications, of a digital correlator implemented in a Xilinx Field Programmable Gate Array is presented. Spread spectrum communications, correlators and Xilinx Field Programmable Gate Arrays are first introduced. A digital correlator is designed and implemented in a Xilinx Field Programmable Gate Array. Cascading correlators to increase the correlation dimensions is next explored. The maximum operating frequency of the digital correlator is calculated and simulated to be 40 Mhz; a significant speed improvement over commercially available correlators such as those from TRW.

DIGITAL CORRELATORS FOR SPREAD SPECTRUM COMMUNICATIONS

by
Raymond B. Chadwick

Thesis submitted to the Faculty of the Graduate School of the New Jersey Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science In Electrical Engineering

1990

APPROVAL PAGE
Digital Correlators for
Spread Spectrum Communications

by
Raymond B. Chadwick

12/20/90

Dr. John Carpinelli, Thesis Advisor
Assistant Professor of Electrical and
Computer Engineering, NJIT

12/20/90

Dr. Anthony Robbi, Committee Member
Associate Professor of Electrical and
Computer Engineering, NJIT

12/20/90

Dr. Sol Rosenstark, Committee Member
Professor of Electrical and Computer
Engineering, NJIT

BIOGRAPHICAL SKETCH

Name: Raymond B. Chadwick

Present Address:

Degree and Date to be Conferred: M.S.E.E., January, 1991

Date of Birth:

Place of Birth:

Collegiate Institutions	Date Attended	Degree	Date of Degree
New Jersey Institute of Tech.	9/88-12/90	MSEE	1/91
New Jersey Institute of Tech.	9/84-5/88	BSEE	5/88

Major: Electrical Engineering

Positions Held:

1/89-Present Kyros Corporation	Digital Design Engineer
6/88-1/89 Siemens Hearing Instruments	Assistant Engineer

ACKNOWLEDGMENT

Appreciation is expressed to Dr. John Carpinelli, Assistant Professor of the Electrical and Computer Engineering Department at the New Jersey Institute of Technology, for his teaching and advisement.

I would also like to acknowledge William Heath, C.E.O. of Kyros Corporation, for his help and use of Kyros' computer facilities.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
CORRELATOR OVERVIEW

1.1 Correlators and Spread Spectrum

Correlators have the ability to detect a certain signal and reject others. The correlators examined in this thesis are those that have found use in Spread Spectrum Communications. In particular, direct sequence spread spectrum systems that use Binary Phase Shift Keying, BPSK, will be considered. Direct-sequence spread spectrum is bandwidth spreading of a data-modulated carrier with a spreading code.

Spread spectrum is a modulation and demodulation technique that can be used to mitigate interference from multiple transmission paths and jammers [1]. Spread spectrum systems use a bandwidth that is much larger than the bandwidth required to transmit the digital information. Specifically, for a communications system to be classified as spread spectrum, it must have the following characteristics. The transmitted signal's bandwidth must be larger than the information bit rate (usually much larger) [1]. The received signal must be demodulated, in part, by correlation with a replica of the spreading code used in the transmission of the spread signal [1].

The amount of signal-to-noise improvement that is achieved by a spread spectrum system is defined as processing gain. A spread spectrum system develops process gain from the bandwidth spreading at the transmitter and the despreading at the receiver. When the received spread spectrum signal is despread, the desired signal collapses to its original bandwidth before spreading. Any signal that does not match the local reference code is spread to at least reference code's bandwidth [2]. The process gain of a spread spectrum system may be approximated by:

Process Gain = Gp = (B.W. RF)/(Information Bit Rate) [2].

It is interesting to note that doubling the RF bandwidth will increase the process gain by only 3 dB.

In a typical spread spectrum system, each data bit is encoded with a spreading code. The encoder may simply exclusive Or the data and the spreading code. The period of 1 bit in the spreading code is defined as a chip. Typical spreading codes are in the range of 15 chips to 10,000 chips in length. Figure 1.1 depicts a typical spread spectrum encoder.

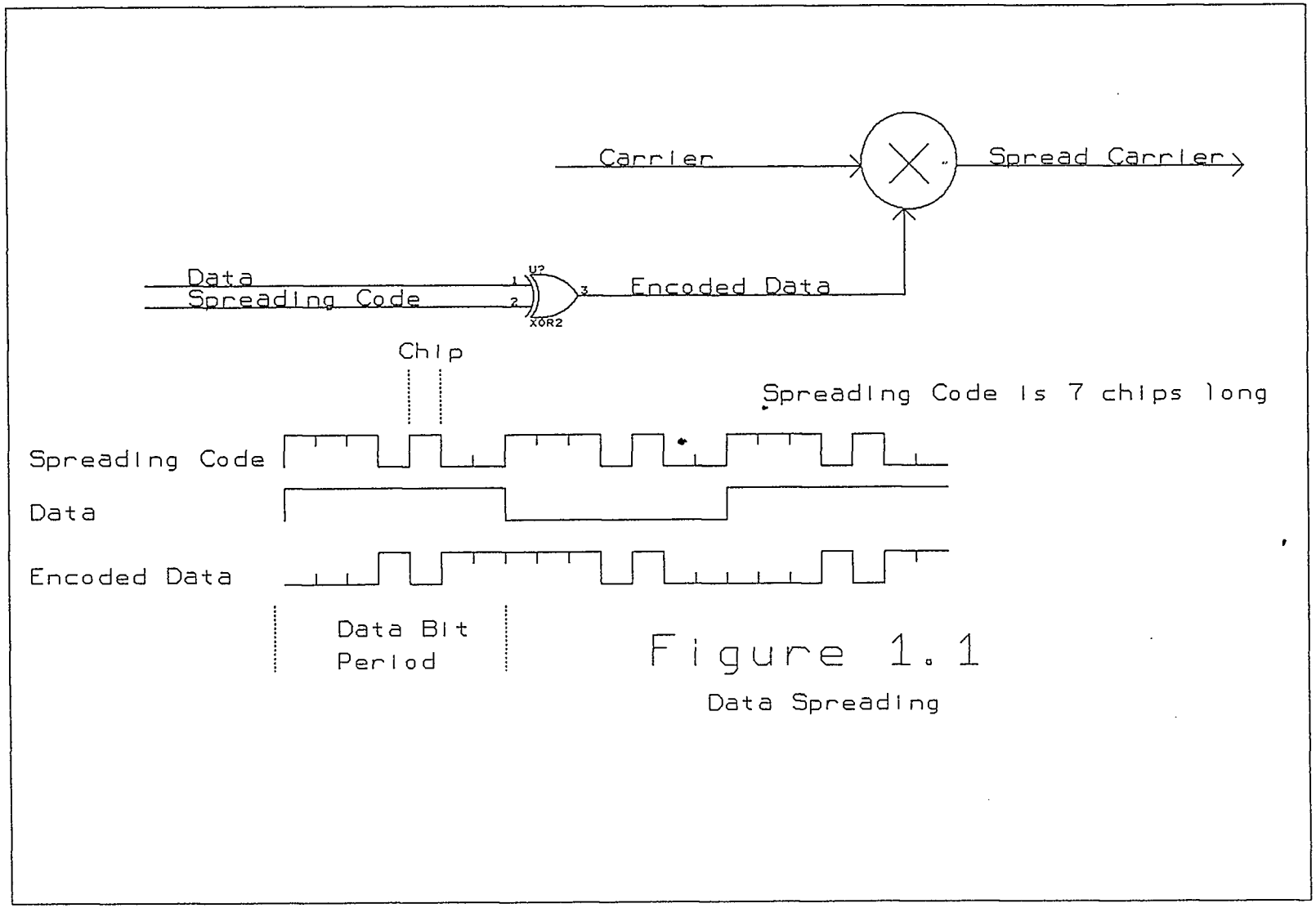


Figure 1.1

Data Spreading

Figure 1.2 depicts a typical Binary Phase Shift Keying, BPSK, spread spectrum transmitter, and figure 1.3 is a typical spread spectrum receiver. Note that the carrier is first modulated by the data and then by the spreading code. A single modulator as shown in figure 1.1 will produce the same analysis. The data modulated signal is expressed by $S_{data}(t)=P\ \cos[W_0t+Od(t)]$. Spreading the signal is accomplished by multiplying $S_{data}(t)$ with a spreading code C(t). The transmitted signal is $S_{trans}(t)=p\ C(t)\ \cos[W_0t+Od(t)]$.

The receiver in figure 1.3 has the despreader at the carrier frequency, or the signal may be mixed down to some intermediate frequency before despreading. A component out of the despreading mixer is P C(t-Td) C(t-Td1) $\cos[W_0t+Od(t-Td)+p]$ where Td is the transmission delay, and Td1 is the receiver's estimate of the transmission delay. If Td=Td1, C(t-Td) C(t-Td1) will be unity, and the output of the despreader is equal to $S_{data}(t)$; S_{data} may then be demodulated for data recovery. Figure 1.4 illustrates the waveforms depicted above, and Figure 1.5 depicts the spectral power density. Figure 1.5 also depicts the spreading of a jammer signal and the despreading of the desired signal.

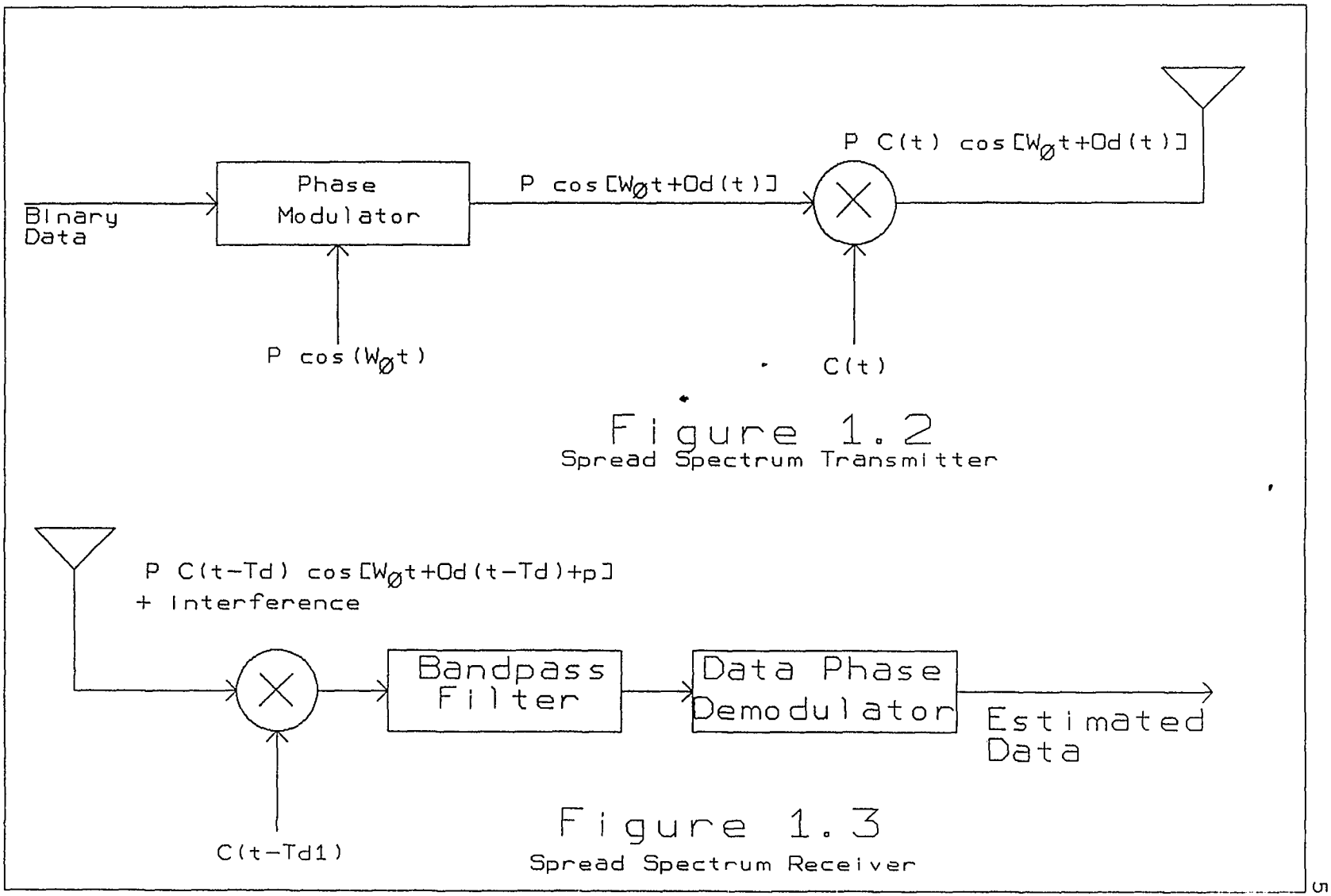


Figure 1.2
Spread Spectrum Transmitter

Figure 1.3
Spread Spectrum Receiver

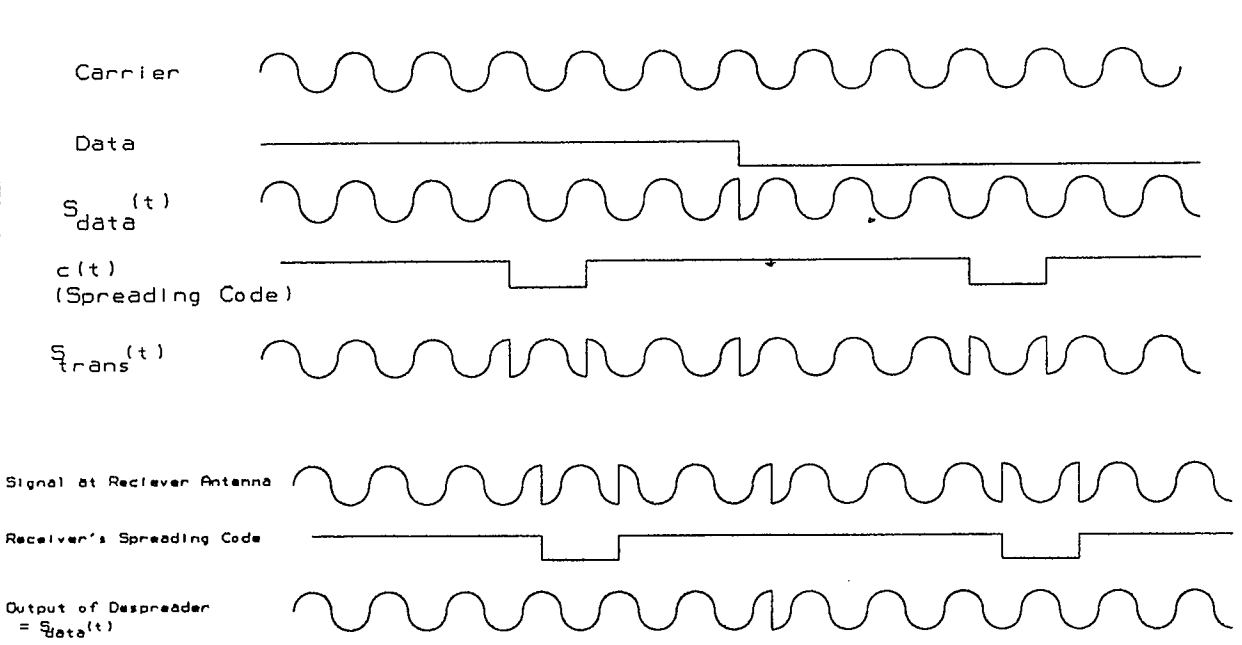


Figure 1.4
Transmitter and Receiver Waveforms.

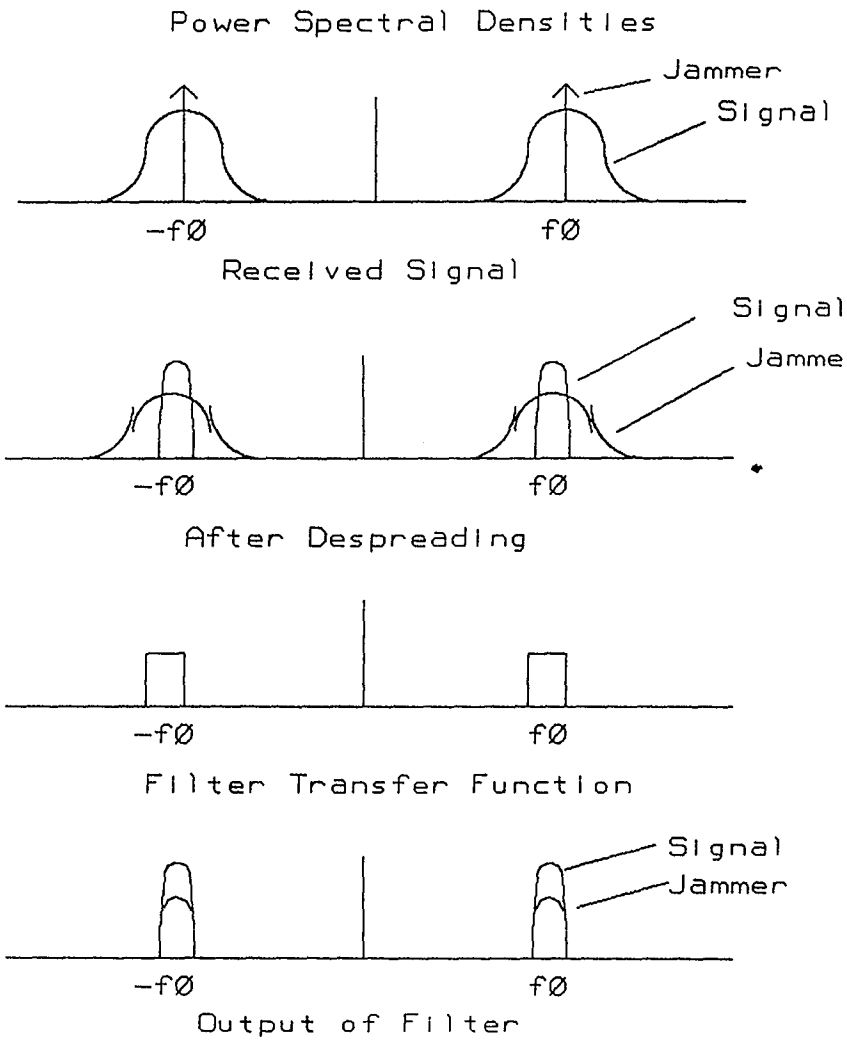


Figure 1.5

Despreading of the desired signal and spreading of a tone jammer.

The despreading operation described above took place at the carrier frequency, or more practically at an intermediate frequency. The despreading may also take place in the base band. The carrier modulated by the data and the spreading code is demodulated to the base band. The recovered signal is, ideally, at the chip rate and the spreading code for a data bit of 0, or the inverted spreading code for a data bit of 1. An analog base-band correlator is shown in figure 1.6. The base-band signal is multiplied by the output of the local spreading code generator. If the base-band signal and the local spreading code are equal and in phase, the output of the multiplier will be a constant over the period of the data bit. The output of the multiplier may then be processed through an integrate and dump filter. The output of the filter may be sampled at the end of the data bit period. The voltage level will indicate how closely the base-band signal and the local spreading code are matched. Figure 1.6 illustrates the despreading of the desired signal and the spreading of undesired signals. Note that the output of the multiplier is a constant for the desired signal, and is chopped up for a different signal. Ideally, the output of the integrate and dump filter will approach n,

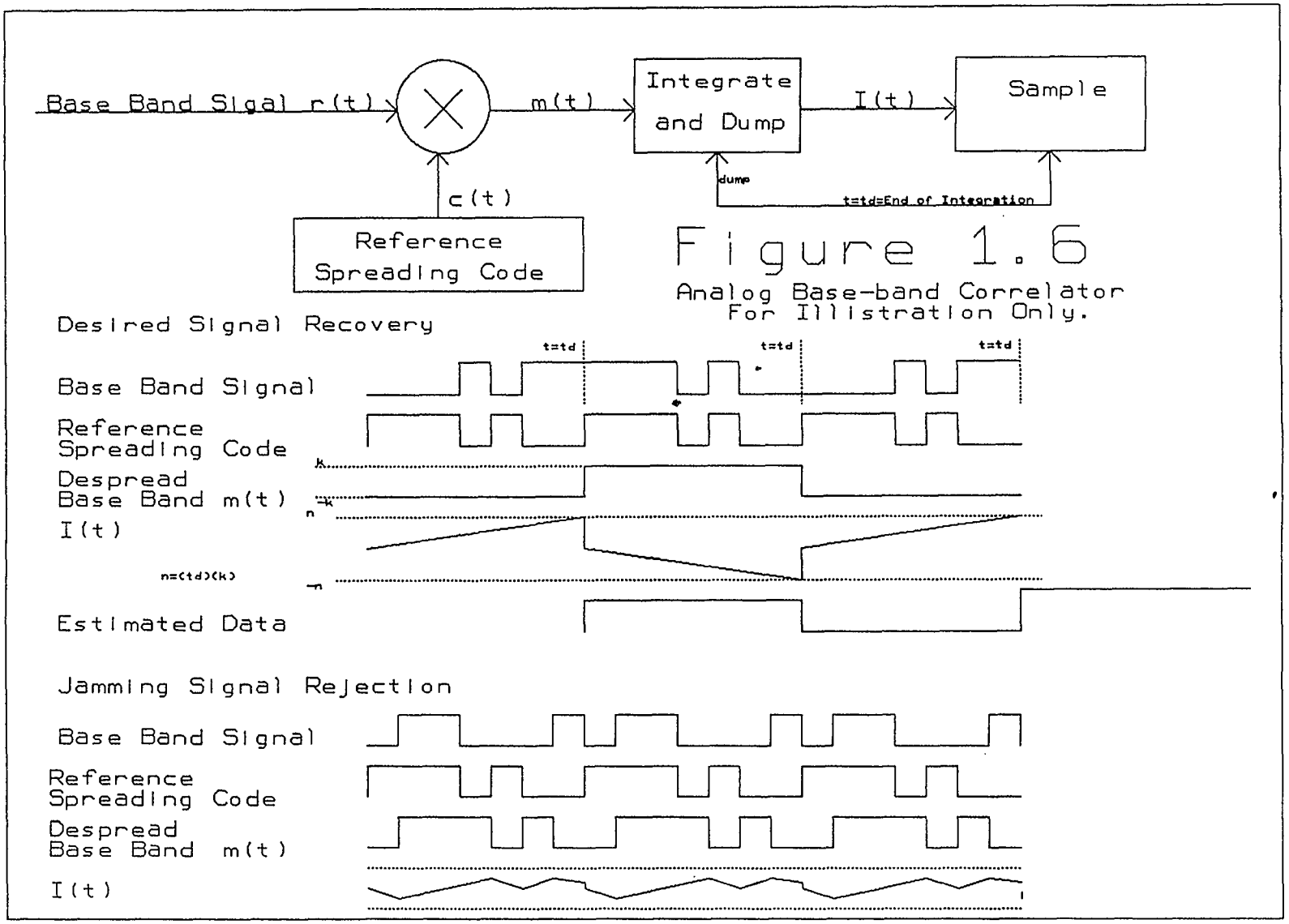


Figure 1.6

Analog Base-band Correlator
For Illistration Only.

as shown in figure 1.6, for the desired signal and zero for a different signal at the end of the data bit period.

The system of figure 1.6 is matched to a certain code, and is said to be a matched filter. The ability of the matched filter to respond to the spreading code and to reject other signals makes it useful in systems where jammers or multiple transmission paths are present, or in Code Division Multiple Access, CDMA, systems. A digital base-band correlator appears in figure 1.7. The multiplier is replaced with an exclusive-or gate, and the integrate and dump filter is replaced with an adder. The despreading and spreading operations are similar, and appear in figure 1.7.

Initial synchronization of the reference spreading code generator in a spread spectrum receiver is a significant problem. The serial search technique is often used to determine the proper phase of the reference spreading code generator. Each spreading code phase is evaluated by attempting to despread the received signal. If the estimated code phase is correct, despreading will occur and be detected. If the code phase is incorrect, the received signal will not be despread, and the

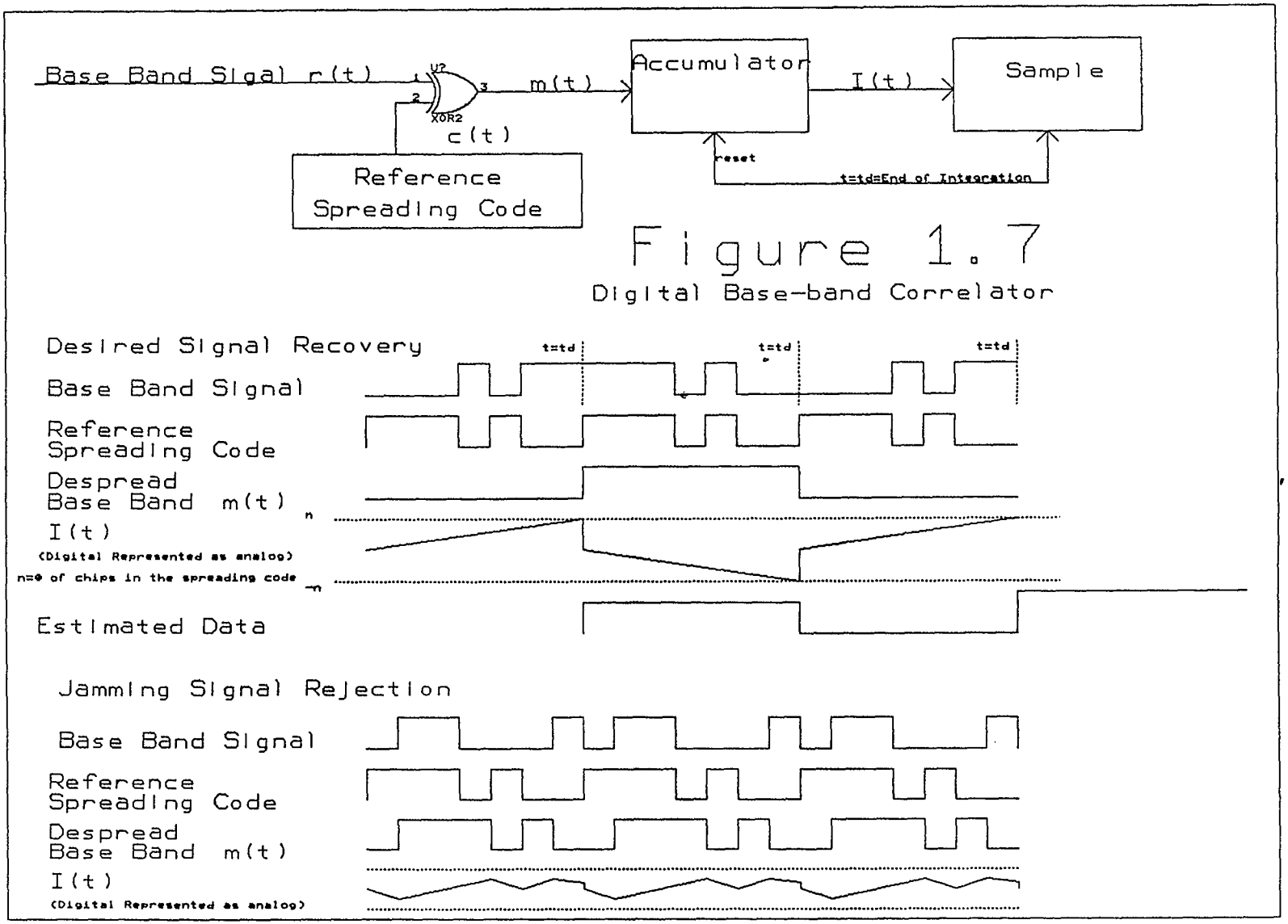


Figure 1.7

Digital Base-band Correlator

reference code generator will be stepped to a new phase. The amount by which the phase of the reference code is stepped is typically 1 chip, 1/2 chip or 1/4 chip. The receiver is searching through code cells, where the span of each cell is equal to the amount by which the reference code is stepped. If the spreading code length is 127 and the phase is stepped by 1/2 chip, there are 254 code cells.

A matched filter that searches many code cells in parallel is shown in figure 1.8. The matched filter, or correlator, is clocked at the chip rate. The base-band signal is clocked into a shift register; the depth of the shift register is equal to the number of chips in the spreading code. The contents of the shift register are compared to the reference code. The number of matches is added, and this is the output of the correlator. If the stepping rate is 1/2 chip, the correlator may be clocked on the rising edge of the clock where 1/2 of the code cells are searched. After these code cells are searched, the correlator is clocked on the falling edge to search the other code cells. Alternatively, two correlators may be used, one on the rising edge and one on the falling edge to search all the code cells in parallel.

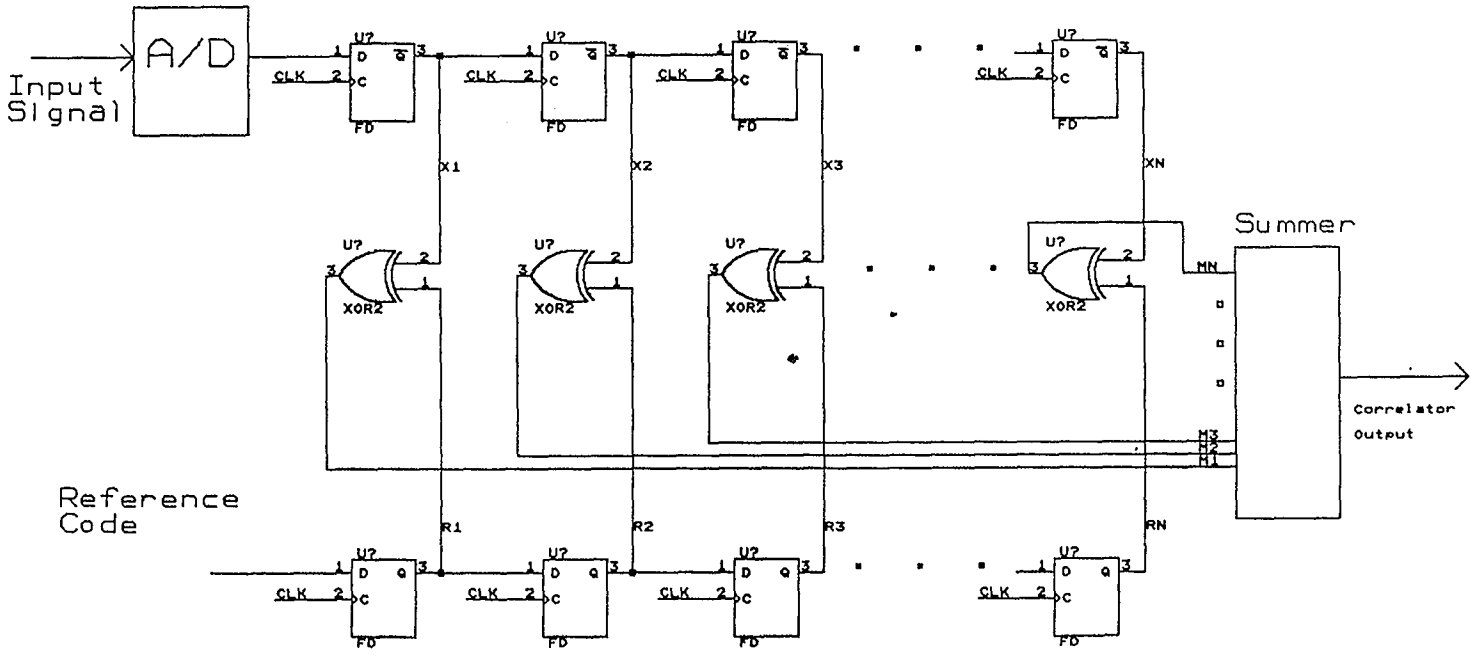


Figure 1.8 Digital Base-band Correlator

CHAPTER 2
XILINX OVERVIEW

2.1 Introduction to Xilinx

Xilinx's Field Programmable Gate Arrays, FPGAs, are similar to other gate arrays; both have an interior of logic and register resources surrounded by Input/Output resources. Xilinx partitions the Field Programmable Gate Array's logic and register resources into a matrix of Configurable Logic Blocks. Interconnection resources are located between the Configurable Logic Blocks and around the I/O blocks; see figure 2.1.

The FPGA is a program driven device; the function of each Configurable Logic Block, I/O block and their interconnections is stored on the FPGA in a static memory. The configuration memory may be loaded from a PROM upon power-up, or it may be programmed by a microprocessor.

The basic memory cell used for the FPGA's configuration memory consists of two CMOS inverters and a pass transistor, as shown in figure 2.2, which is used to write or read cell data. The cell is written-to only during configuration and is read only during a readback check of the configuration. During normal operation, the pass transistor is off, and it does not affect the stability of the memory cell [3].

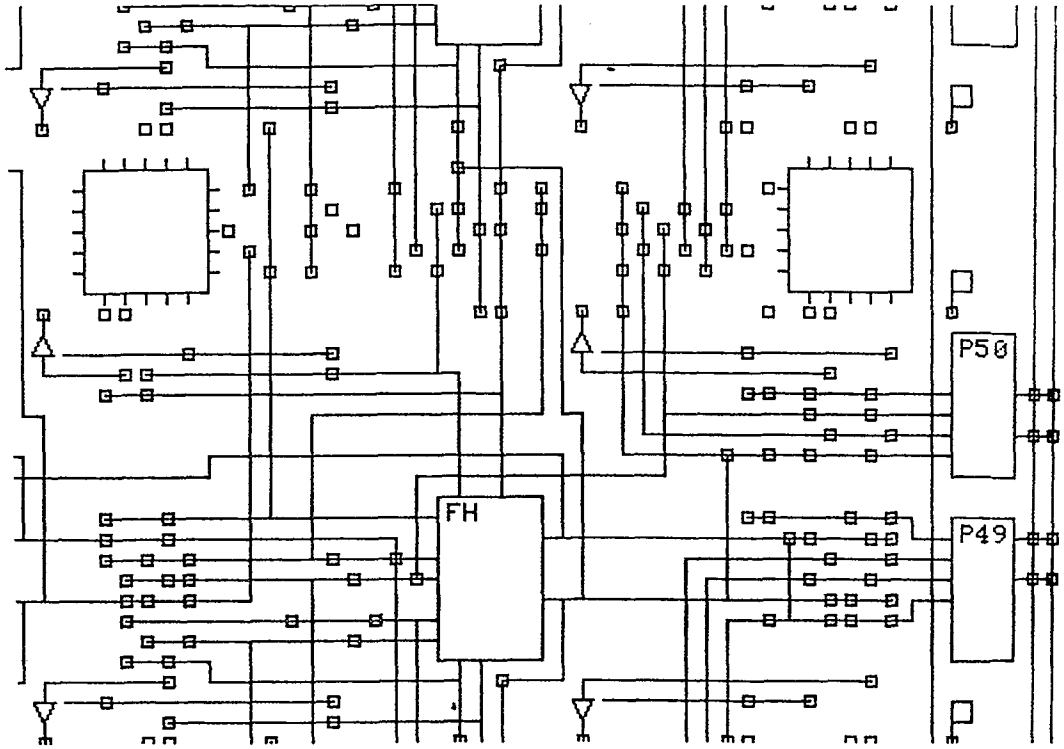


Figure 2.1. CLBs, IOBs AND Interconnect.

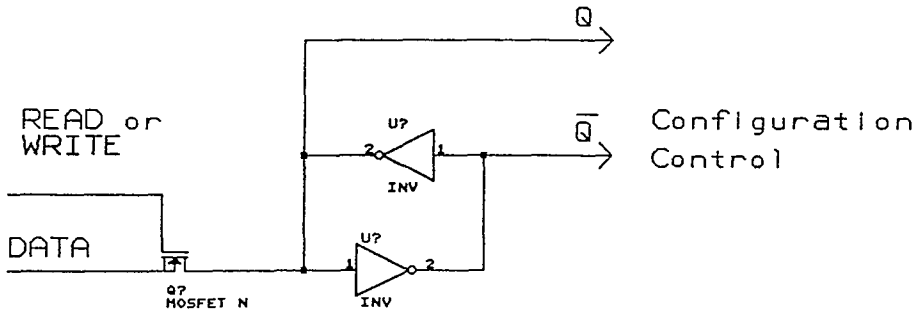


Figure 2.2
Static Memory Cell

Each user-configurable I/O Block, IOB, has a registered and a direct input path from its corresponding package pin. Each IOB has a programmable three-state output buffer which may be driven by a registered or direct output signal. Each input circuit also provides input clamping diodes to provide electro-static protection and circuitry to inhibit latch-up. An IOB schematic appears in figure 2.3.

The array of Configurable Logic Blocks, CLBs, are the elements used to construct user logic. The XACT development system is used to compile the configuration data which defines the interconnection and configuration of each CLB. Each CLB has a logic section and two flip-flops. The CLB has five logic inputs, an asynchronous reset and a clock enable. All inputs may be driven by the interconnection resources. Each CLB has two outputs that can drive interconnection resources. Figure 2.4 is a schematic diagram for each CLB; note that the multiplexers drawn as trapezoids and the combinatorial function block are controlled by the configuration memory.

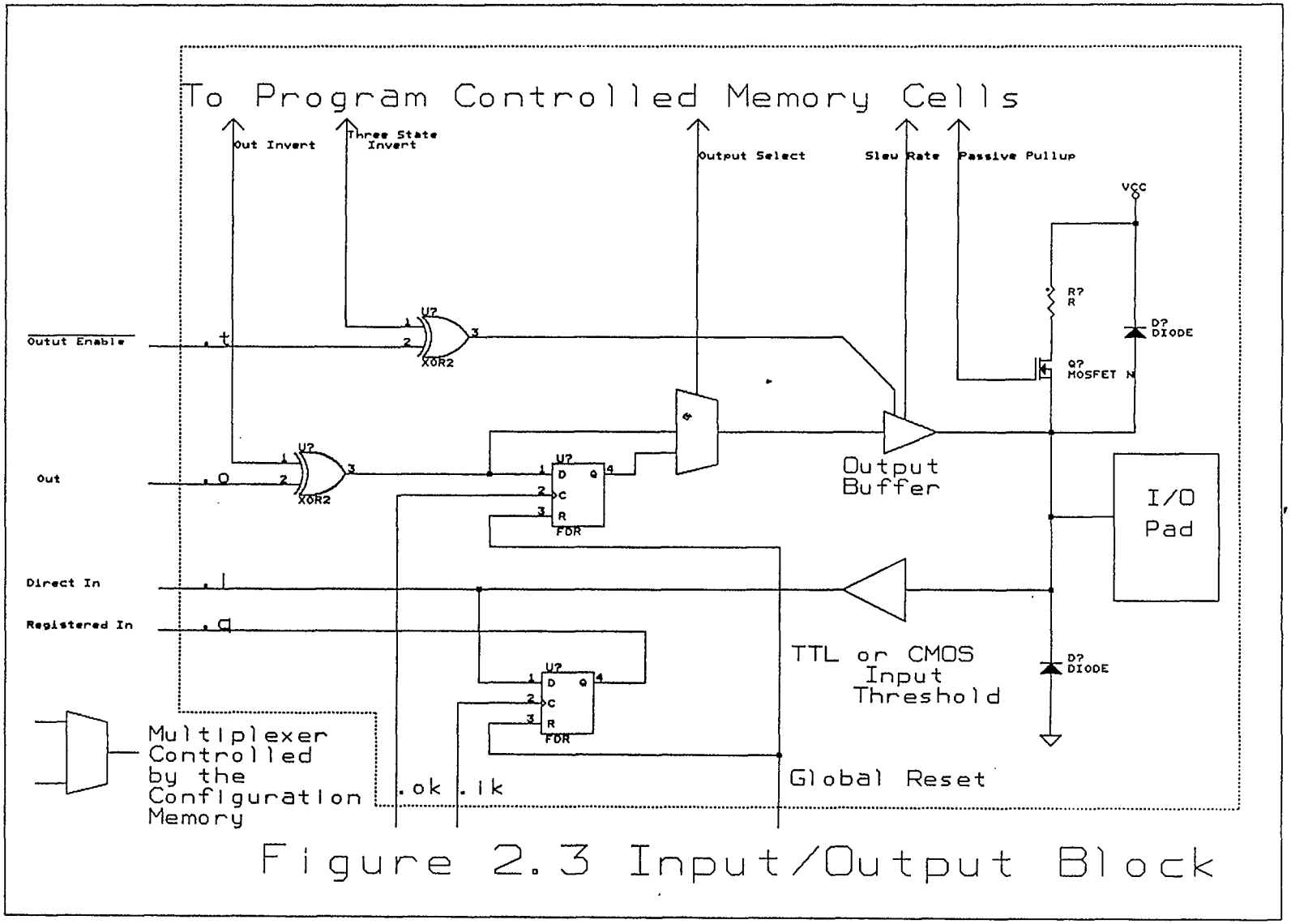


Figure 2.3 Input/Output Block

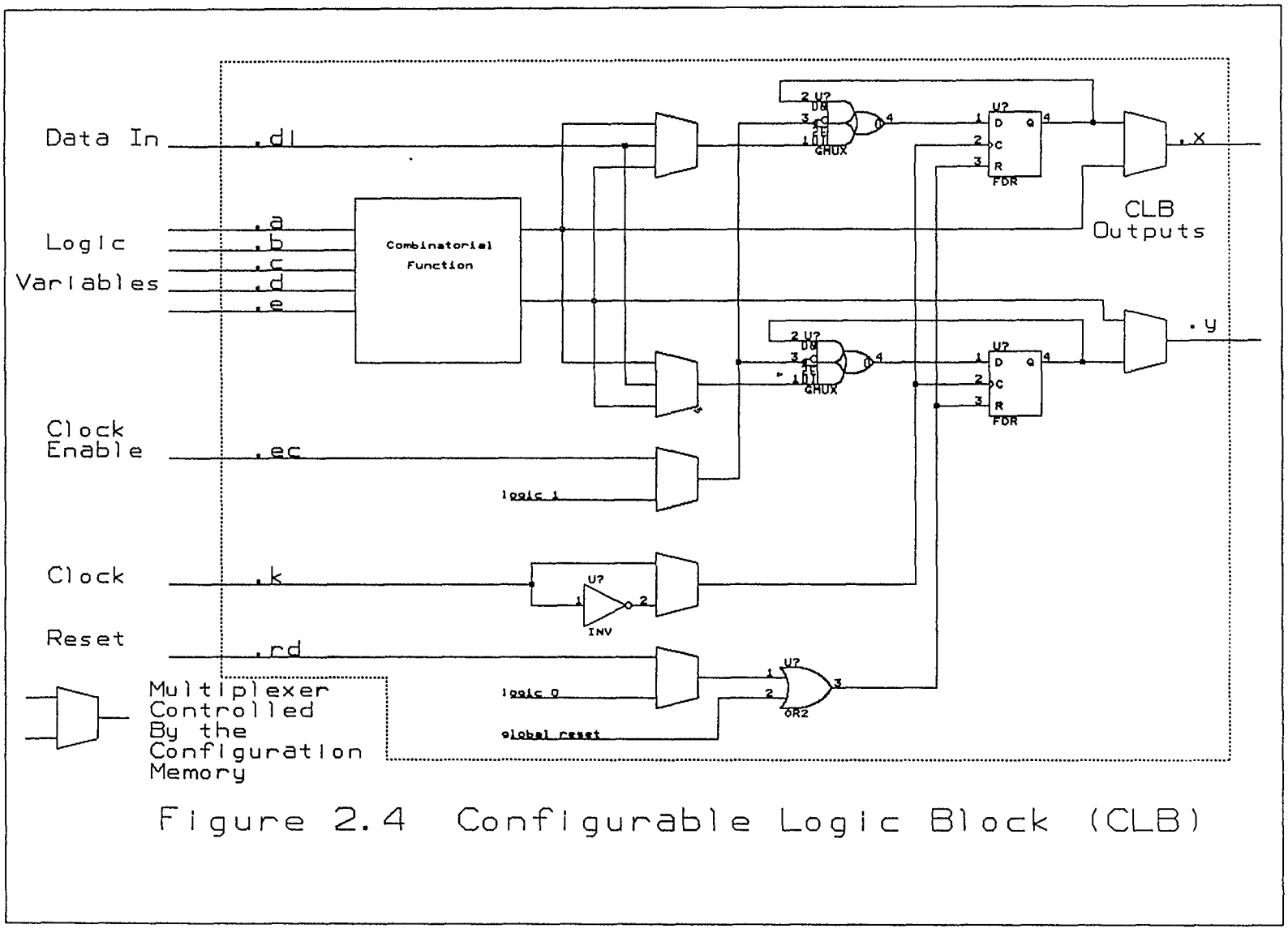


Figure 2.4 Configurable Logic Block (CLB)

The interconnection resources in the Field Programmable Gate Array are programmable, and provide routing paths to connect inputs and outputs of the CLBs and I/O blocks. The interconnect consists of a 2 layer metal grid between the CLBs and I/O blocks. Pass transistors form programmable interconnect points and switching matrices that can connect or isolate metal segments.

2.2 Xilinx Development Systems

The XACT Design Editor is a complete design and development system for implementing designs in Xilinx Field Programmable Gate Arrays. Definition of Configurable Logic Blocks and Input/Output blocks and their interconnection is performed with a menu driven graphics editor. An automatic router is available in the XACT editor or the interconnections may be manually routed by turning on pass transistors and switching matrices.

A companion program, Automatic Place and Route, can rearrange the placement of CLBs and IOBs to optimize the speed of the interconnect. The Automatic Place and Route, APR, does not change the function of the circuit. The APR program then routes the design.

Translators connect the XACT program with simulators and schematic capture programs. Designs may be entered using a schematic capture program and a special library of components supplied be Xilinx. A translator is then used to generate a file compatible with the XACT program. The design is then routed by the user with XACT or the APR. Another translator converts XACT design files to a format compatible with several simulators. The functionality and timing of the Field Programmable Gate Array can then be simulated.

The XACT program translates the FPGA design into a bit stream. The bit stream is the configuration data that programs the FPGA. The XACT program can serially load the FPGA with the bit stream via a special cable that connects the printer port of the host computer to the FPGA; or the XACT program can burn a PROM that contains the bit stream. The PROM can be wired to the FPGA so that the bit stream is loaded on power-up.

CHAPTER 3
CORRELATOR DESIGN

3.1 Correlator Design

A digital correlator as described earlier, one that searches through multiple code cells in parallel, was implemented in a Xilinx Field Programmable Gate Array. The schematics for the design appears in figures 3.1 - 3.13, and the placed and routed Xilinx design appears in the appendix.

The design consists of two independent correlators, correlators a and b, of dimension n X m, where n=15 and m=1. The term correlator will refer to one 15 X 1 correlator implemented in the Xilinx FPGA. The design correlates a signal that is up to 15 chips long and quantized to 1 bit. Longer codes or signals quantized to more bits or both may be correlated by cascading correlators in series, parallel or both. The correlator's dimension n may also be decreased to a minimum of 7. Each 15 X 1 correlator consists of a 15 bit shift-register used to store the previous 15 bits of the sampled signal, a matching section where each bit in the shift-register is compared to its corresponding reference code bit, a pipelined adder that adds the number of matches and support circuitry for cascading correlators.

The shift register and matching sections are straight forward. Alternate registers in the shift-register have enable inputs. If the enable is low, the register is 'MUXed' out of the shift-register. The first enable bit, E0, 'MUXes' four registers, E1 'MUXes' two registers, E2 'MUXes' one register and E3 'MUXes' one register. The shift-register may be varied from 7 to 15 bits. The matching section simply exclusive ORs the reference code, R0 - RE, with the shift register's contents on a bit by bit basis and registers the result. Alternate matching sections also have enable bits. If the exclusive OR's corresponding register in the shift-register is not enabled, the exclusive OR's output is masked to a low. The correlator's length may be varied from 7 to 15 using the enable bits.

The correlator 15 X 1 schematic, figure 3.3, descends into several sheets. Each sheet corresponds to the configuration of a CLB in the Xilinx. The Add3 sheet, figure 3.8, implements a 3-bit adder in 1 CLB. Add2, figure 3.9, is a 2-bit adder. Add3b, figure 3.10, is a 3-bit adder with carry look-ahead. Add14 and Add8 are also carry look-ahead adders that are described later. The CLB's name, in figure 3.3, appears at the top of the sheet in brackets. Each CLB in the pipelined adder begins with

an S; the next two letters correspond to the location of the CLB in the schematic. The final placement of the CLBs was changed by the APR program. The first 3 letters of the net names in the pipelined adder are the name of the CLB that sources the net. The following number describes the weight of the bit generated by the adder, i.e. 1 for binary 1's place and 2 for binary 2's place.

The pipelined adder has four sections. The first section contains 3-bit adders. The adders generate five 1's place and five 2's place results. The next stage adds the five 1's with a 3-bit adder and a 2-bit adder. The 2's are also added together with a 3-bit adder and a 2-bit adder.

The third stage adds the two 4's place results, SDD4 and SDG4, with a 2-bit adder. Add3b adds the two 1's, SCB1 and SCE1, to generate the final result for the 1's place. It also has carry look-ahead from the addition of the two 1's and adds the result to SDF2.

The results of the final stage are sent to IOBs for output to the FPGA's pins. Block Add2a adds the last 2's together to compute the final result for the 2's place. A carry look-ahead exists in Add14 and Add8. Add14, figure 3.12, registers the 1's place result computed in the

stage above; it also produces the final 4's place result using carry look-ahead. Add8, figure 3.13, computes the final 8's place result with carry look-ahead from the 2's place and the 4's place.

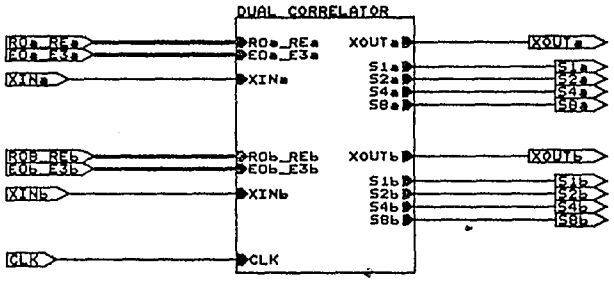


Figure 3.1

Dual Correlator - Root Level

R. CHADWICK NJIT		
Title: DUAL CORRELATOR - ROOT LEVEL		
Size: B	Document Number	REV: 0.1
Date: November 28, 1990	Sheet of	

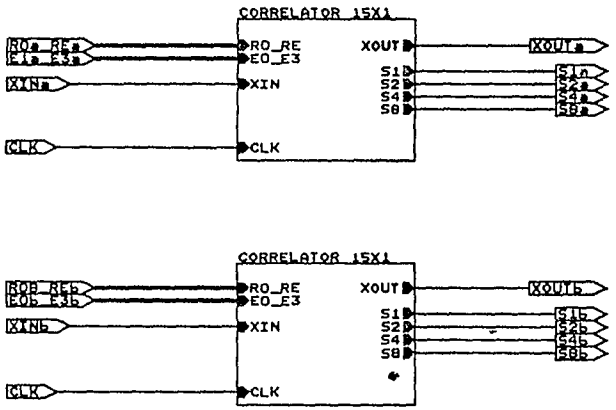


Figure 3.2

Dual Correlator

R. CHADWICK NJIT		
Title: DUAL CORRELATOR		
Size: B	Document Number	REV: 0.1
Date: November 28, 1990	Sheet of	

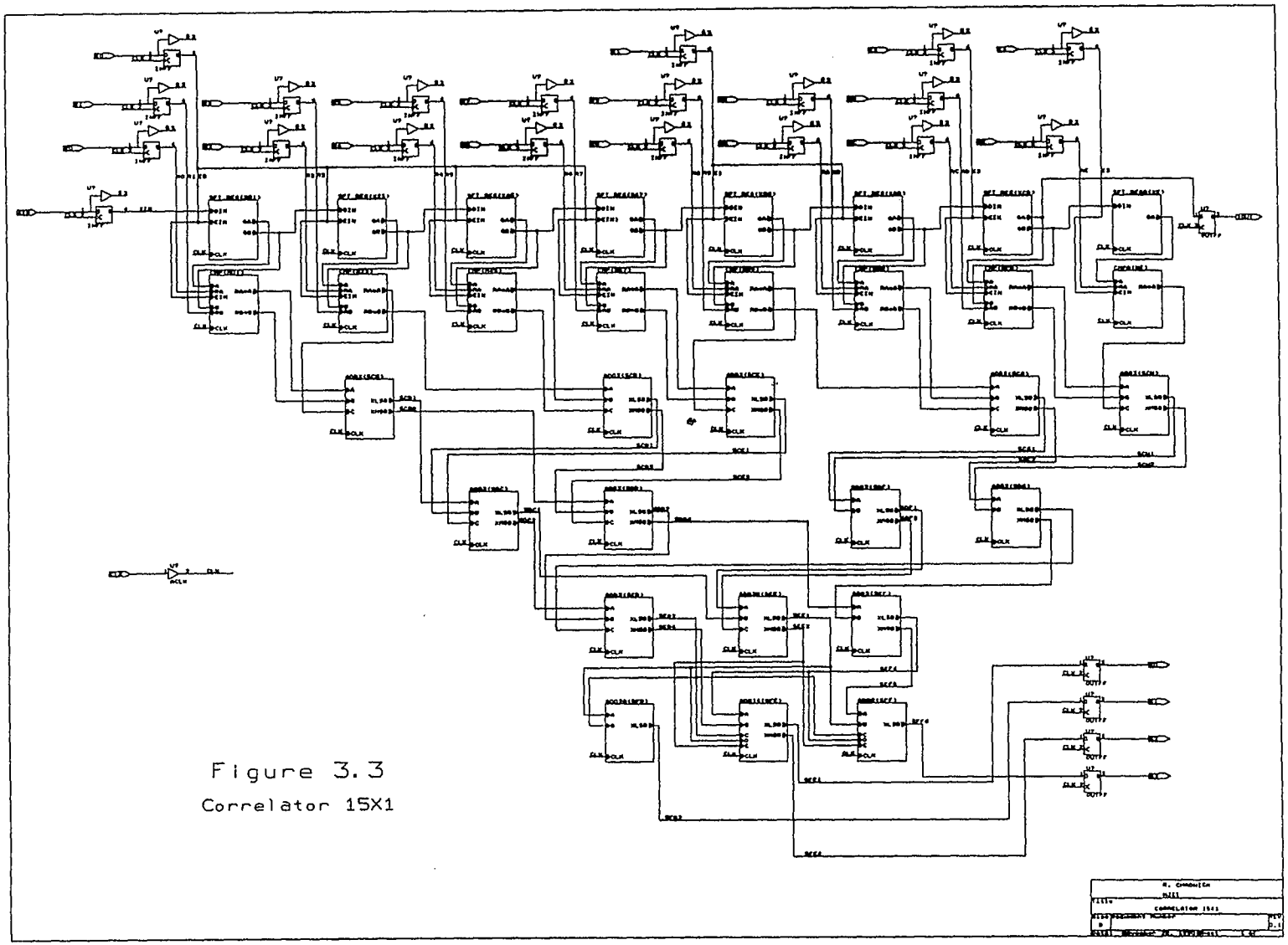


Figure 3.3

Correlator 15X1

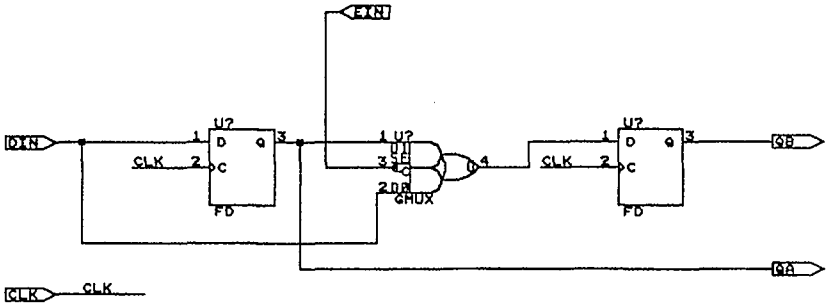


Figure 3.4
Shift-Register

Sheet Name: SFT_REG

Figure 3.5
Shift-Resister (Last Stage)

Sheet Name: SFT_REGA

R. CHADWICK NJIT		
Title: SFT_REG / SFT_REGA		
Size: B	Document Number	REV: 0.1
Date: November 28, 1990	Sheet of	

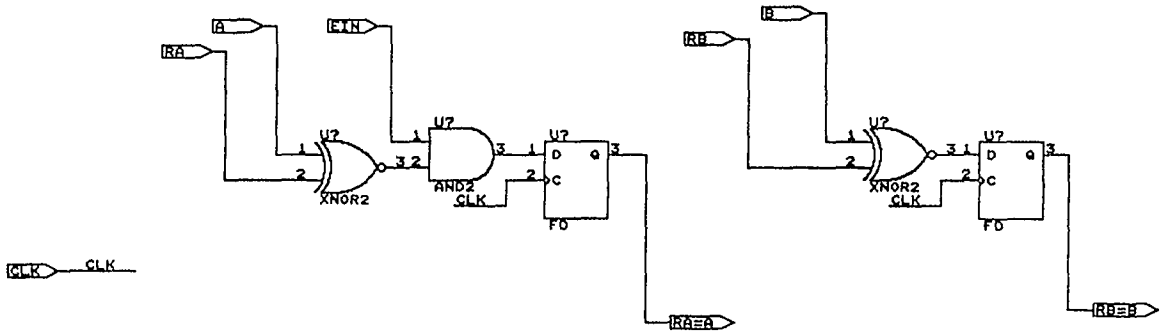


Figure 3.6
Comparators

Sheet Name: CMP

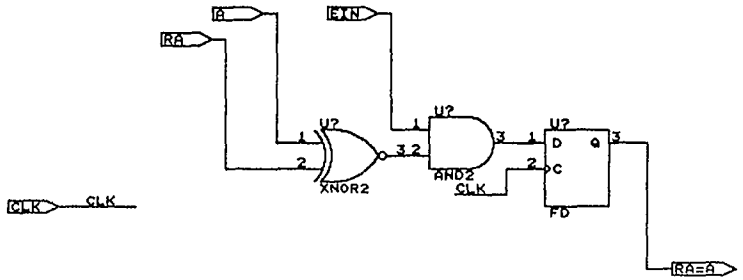


Figure 3.7
Comparator (Last Stage)

Sheet Name: CMPA

	R. CHADWICK NJIT	
Title	CMP / CMPA	
Size B	Document Number	REV 0.1
Date: November 29, 1990	Sheet of	

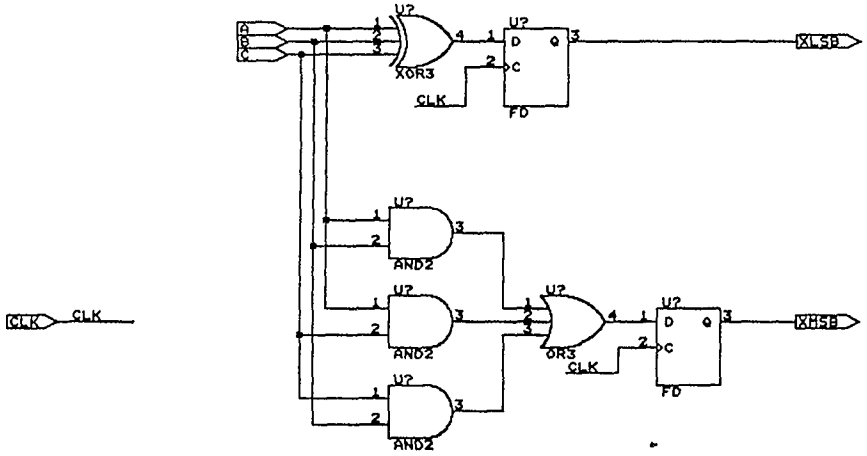


Figure 3.8

3-Bit Adder

Sheet Name: ADD3

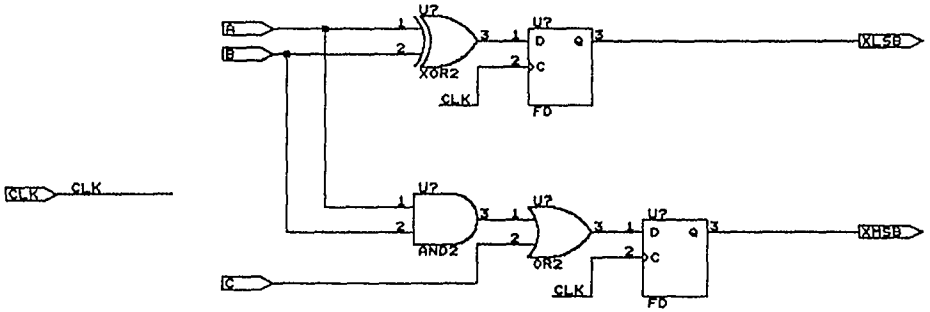


Figure 3.9

3-Bit Adder with Carry Look-Ahead

Sheet Name: ADD3B

R. CHADWICK NJIT		
Title: ADD3 / ADD3B		
Size: B	Document Number	REV: 0.1
Date: November 28, 1990	Sheet of	

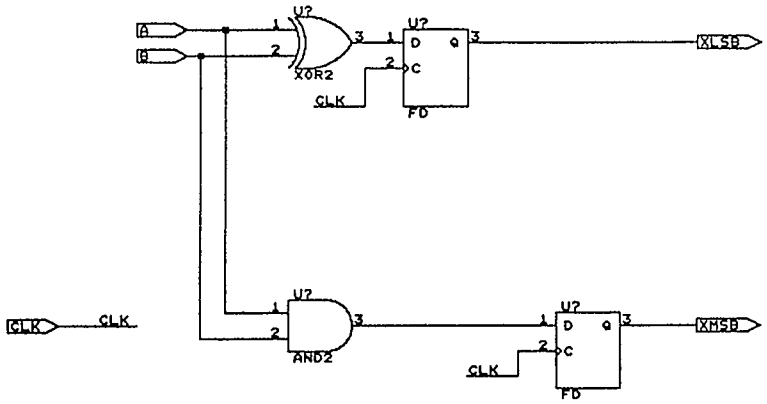


Figure 3.10
2-Bit Adder

Sheet Name: ADD2

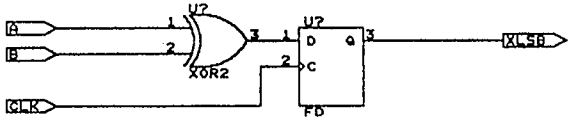


Figure 3.11
Half-Adder

Sheet Name: ADD2A

R. CHADWICK NJIT		
Title: ADD2 / ADD2A		
Size: B	Document Number	REV: 0.1
Date: November 20, 1990	Sheet of	

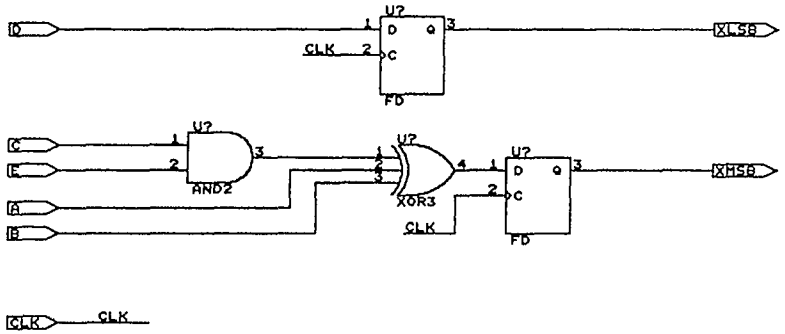


Figure 3.12
Adder with Carry Look-Ahead

Sheet Name: ADD14

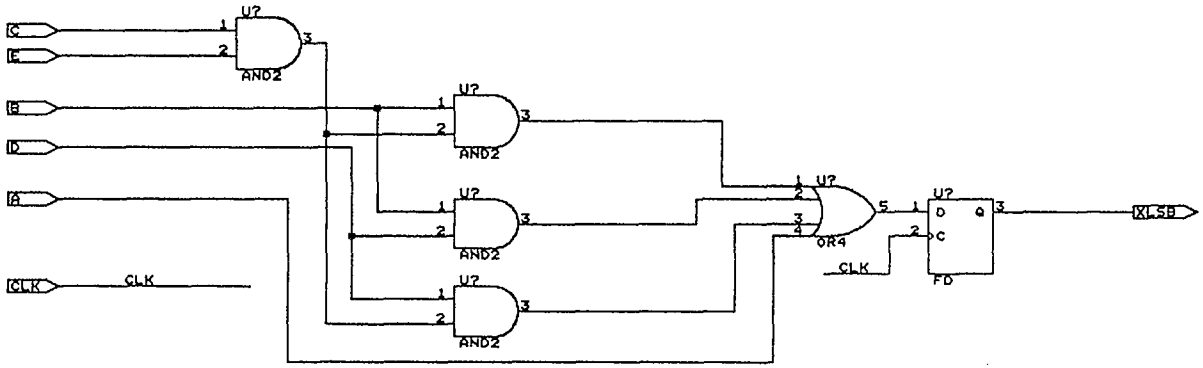


Figure 3.13
Adder with Carry Look-Ahead

Sheet Name: ADD8

	R. CHADWICK NJIT	
Title	ADD14 / ADD8	
Size B	Document Number	REV 0.1
Date: November 28, 1990	Sheet of	

3.2 Implementation in Xilinx

The circuit shown in figures 3.1 - 3.13 was implemented in a Xilinx 3020PC68-70 FPGA. The 3020 consists of an 8 X 8 matrix of Configurable Logic Blocks and 64 user I/Os in a 68 pin PLCC package. The 3020 is estimated by Xilinx to contain 2000 NAND gate equivalents. The -70 is a speed grade which indicates that the maximum toggle frequency of an internal CLB flip-flop is 70 Mhz. A -100 speed grade is available at this time, but the part is not widely available, and its cost is at a premium. Hence, a -70 part was chosen.

The correlator schematic shown in figures 3.1 - 3.13 was captured using Orcad STD and an X3000 series library supplied by Xilinx. The schematic could have been translated into a file compatible with the XACT program. The design was entered by hand into XACT because it was a regular design with many identical blocks. With the 'hand translation' it is easier to control the partitioning of the logic into CLBs. One complete correlator, the a correlator, was entered, and then the design was duplicated to create the b correlator within the XACT program. The design was then run through the APR program several times. As will be shown later, the input and output pads of the Xilinx have

a maximum frequency of operation, the maximum external operating frequency, that is fixed. The maximum internal frequency, the maximum frequency at which the CLBs with routing delays will still operate properly, is dependent upon the placement of CLBs, IOBs and the routing interconnections. The APR program is pseudo random; the placement of CLBs is partly determined by a random number generator. The APR program was run until a design was created with an internal operating frequency within a few percent of the fixed external operating frequency. The final placed and routed design appears in figure 3.14. The enable nets, E0, E1, E2 and E3, and accompanying logic were exempt from meeting the above criteria. In typical spread spectrum systems, the correlator is usually a fixed length, and the enable bits would be set at system initialization.

3.3 Maximum Operating Frequency

The maximum operating frequency of the Xilinx correlator was determined by calculating the maximum operating frequency for several different parts of the design. The minimum frequency calculated then determines the maximum operating frequency of the device.

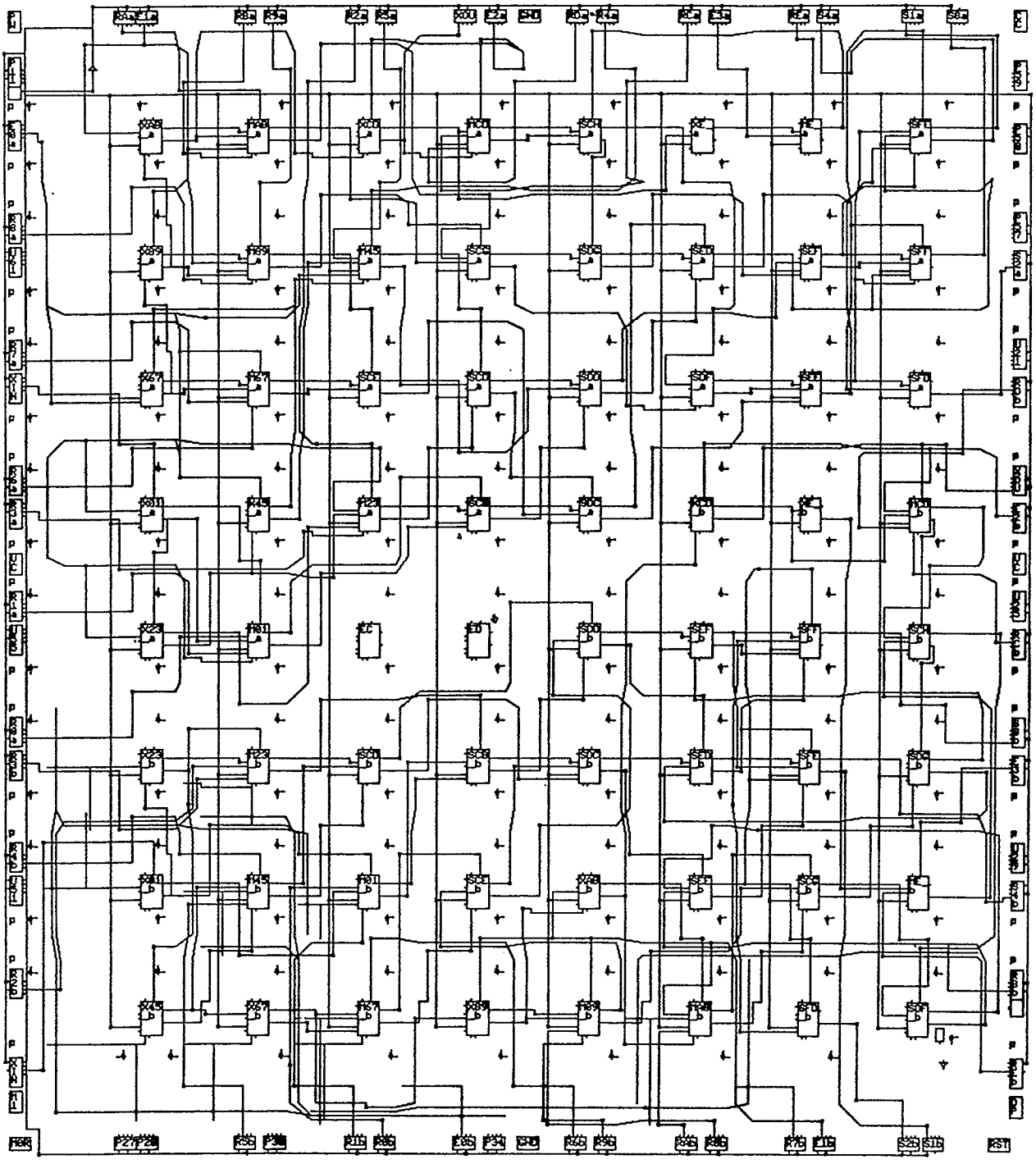
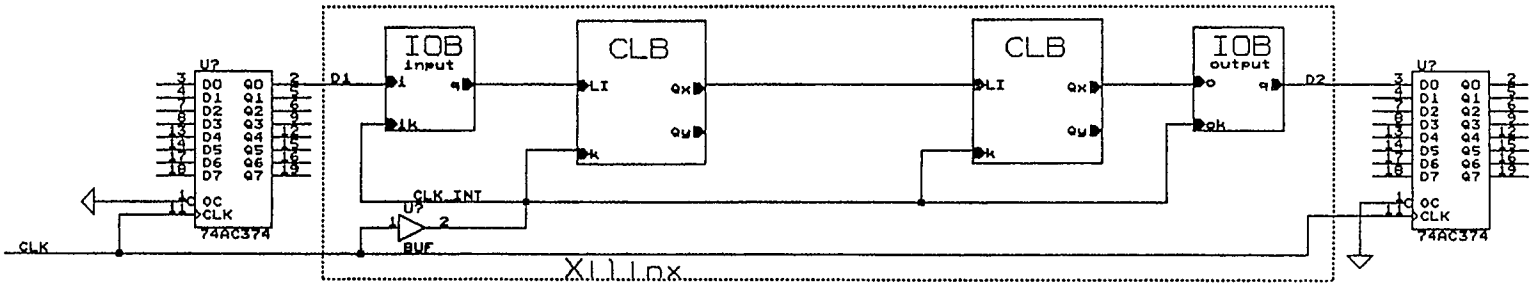


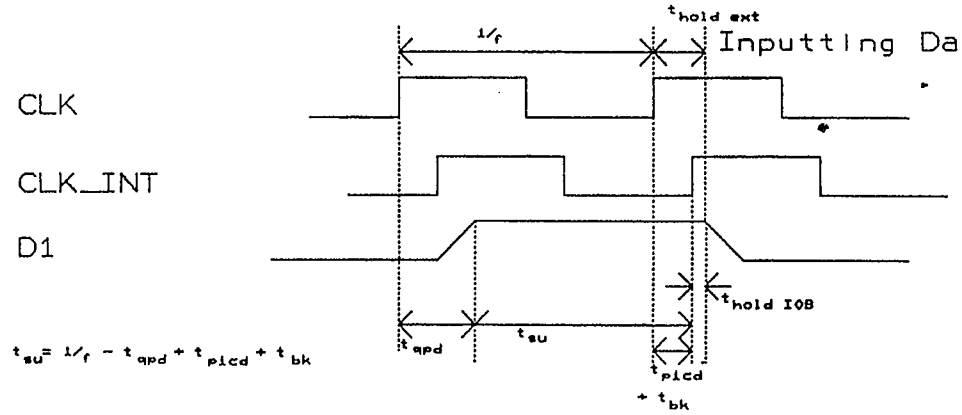
Figure 3.14. The final placed and routed design.

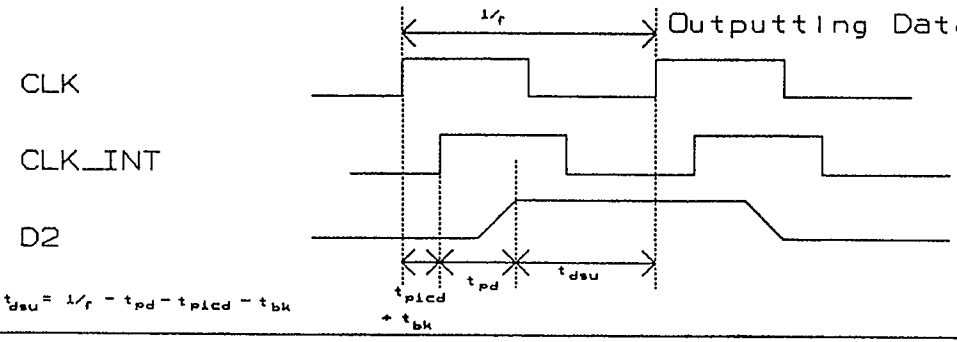
The propagation delay from the clock of an IOB(ik) or CLB(k) to its output is 8ns. The time through a CLB's combinatorial logic and the time needed to set up its register is 8ns. The set up time for an output IOB(o) is 10ns. See Figure 3.15. The propagation delays are over the operating conditions shown in figure 3.16. Since every CLB and IOB has a registered output, the maximum internal operating frequency will be determined by the longest delay from the clock to an IOB's or CLB's output, through the routing and then the set-up at the net's load, a CLB or IOB.



Xilinx and 74AC374 I/O delays

Figure 3.15





$T_{ambient}$ = 0 degrees Celsius to 75 degrees Celsius
Supply Voltage = 4.5 to 5.5 volts.

Figure 3.16 Operating Temperatures and Voltage supply regulation for a Xilinx XC3020PC68-70 Commercial, a Gazelle GA22V10 and a 74AC374.

The longest net delay from a CLB(Qx or Qy) to an output IOB was found to be net _bSFE1 or 7ns. The Longest net delay from an input, IOB or CLB to a load CLB was 9ns; _bR7 is one such net. Note that the enable nets were not considered here for the reasons stated previously.

The maximum frequency that net _bSFE1 can run is:

$$f1_{max} = 1/(t_{prop\ CLB} + \text{net delay} + t_{set\text{-}up\ output\ IOB}$$
$$= 1/(8ns + 7ns + 10ns) = 40\ Mhz.$$

The maximum frequency that _bR7 can support is:

$$f2_{max} = 1/(t_{prop\ IOB} + \text{net delay} + t_{set\text{-}up\ CLB})$$
$$= 1/(8ns + 9ns + 8ns) = 40\ Mhz.$$

The maximum internal operating frequency is 40 Mhz.

The maximum external operating frequency was calculated by assuming a register was driving the input IOBs and was the load on an output IOB; see figure 3.16.

The following parameters were defined:

t_{picd} Xilinx pad to clock buffer input.

t_{bk} Xilinx clock buffer input to any clock.

t_{su} Xilinx pad to clock(ik) set up.

t_{pd} Xilinx clock(ok) to pad propagation delay.

t_{dsu} Register D set up.

t_{qpd} Register clock to Q delay.

From figure 3.16, the maximum frequency for outputting data from an IOB to the register is:

$$f3_{max} = 1/(t_{dsuMin} + t_{pdMax} + t_{picdMax} + t_{bkMax})$$

$$= 1/(t_{dsuMin} + 13ns + 3ns + 6ns) < 45 \text{ Mhz}.$$

From figure 3.16, the maximum operating frequency for inputting data from the register to an IOB is:

$$f4_{max} = 1/(t_{qpdMax} + t_{suMin} - t_{pidcMin} - t_{bkMin})$$

No minimum values for t_{pdic} or t_{bk} are guaranteed by Xilinx. Xilinx does state, but does not guarantee, a minimum value of 5ns for $t_{pidc} + t_{bk}$ when t_{su} is at its maximum value. If $t_{pidcMin} + t_{bkMin}$ is 5ns:

$$f4_{max\ optimistic} = 1/(t_{qpdMax} + 20ns - 5ns) < 66 \text{ Mhz}.$$

If $t_{pidcMin} + t_{bkMin}$ is set to zero:

$$f4_{max\ conservative} = 1/(t_{qpdMax} + 20ns) < 50\text{Mhz}.$$

If a very fast register is used, the calculated maximum operating frequency would be 40 Mhz. If a GA22V10, a high-performance Programmable Logic Device from Gazelle Microcircuits, is used as a register:

$f3_{max\ GA22V10} = 1/(3ns + 13ns + 3ns + 6ns) = 40$ Mhz.

$f4_{max\ conservative\ GA22V10} = 1/(5ns + 20ns) = 40$ Mhz.

If a 74AC374 is the register used, f3 or the conservative value for f4 would limit the operating frequency.

$f3_{max\ 74AC374} = 1/(4.5ns + 13ns + 3ns + 6ns) = 37$ Mhz.

$f4_{max\ conservative\ 74AC374} = 1/(10ns + 20ns) = 33$ Mhz.

As can be seen in figure 3.16, it is important to consider the case were $t_{picd} + t_{bk} > t_{qpd}$. The hold time for the input IOB is negative. The input IOB has a delay between the package pin and its register. The input set-up time specified for each IOB is this delay plus the register's set-up time. A negative hold time, $t_{hold\ IOB}$, is allowable. Xilinx states that $t_{hold\ ext}$, as shown in figure 3.16, may be zero, and not violate the hold time requirement on the input IOB's register.

The maximum operating frequency for the device was estimated at 40 Mhz, assuming the set-up time on the input IOBs is met. A maximum system operating frequency was calculated to be 40 Mhz. Both frequencies were calculated with the operating conditions listed in figure 3.15.

3.4 Cascading Correlators

Longer codes may be correlated by cascading correlators in series. Figure 3.17 illustrates two correlators in series; the Xout from the first correlator is connected to the Xin of the second. The cascading effectively forms one long, continuous shift-register. Note that the Xout output does not come from the last stage of correlator a's shift-register; it comes two stages earlier because the output IOB of this correlator and the input IOB of the next both contain a register. The pipelined adder in the first correlator adds the number of code matches in the first 15 stages of correlation, and correlator b adds the second 15 stages. The two results must be added externally to form the final result.

There is a quantization error when the signal attempting to be correlated is quantized to one bit. The signal may be quantized to several bits to reduce this error, see figure 3.18. The range of the A/D is from 0 to .11 binary, .75 decimal. Each bit from the A/D is correlated to the reference code. The results are then added after each bit is scaled. The second bit, D1, is

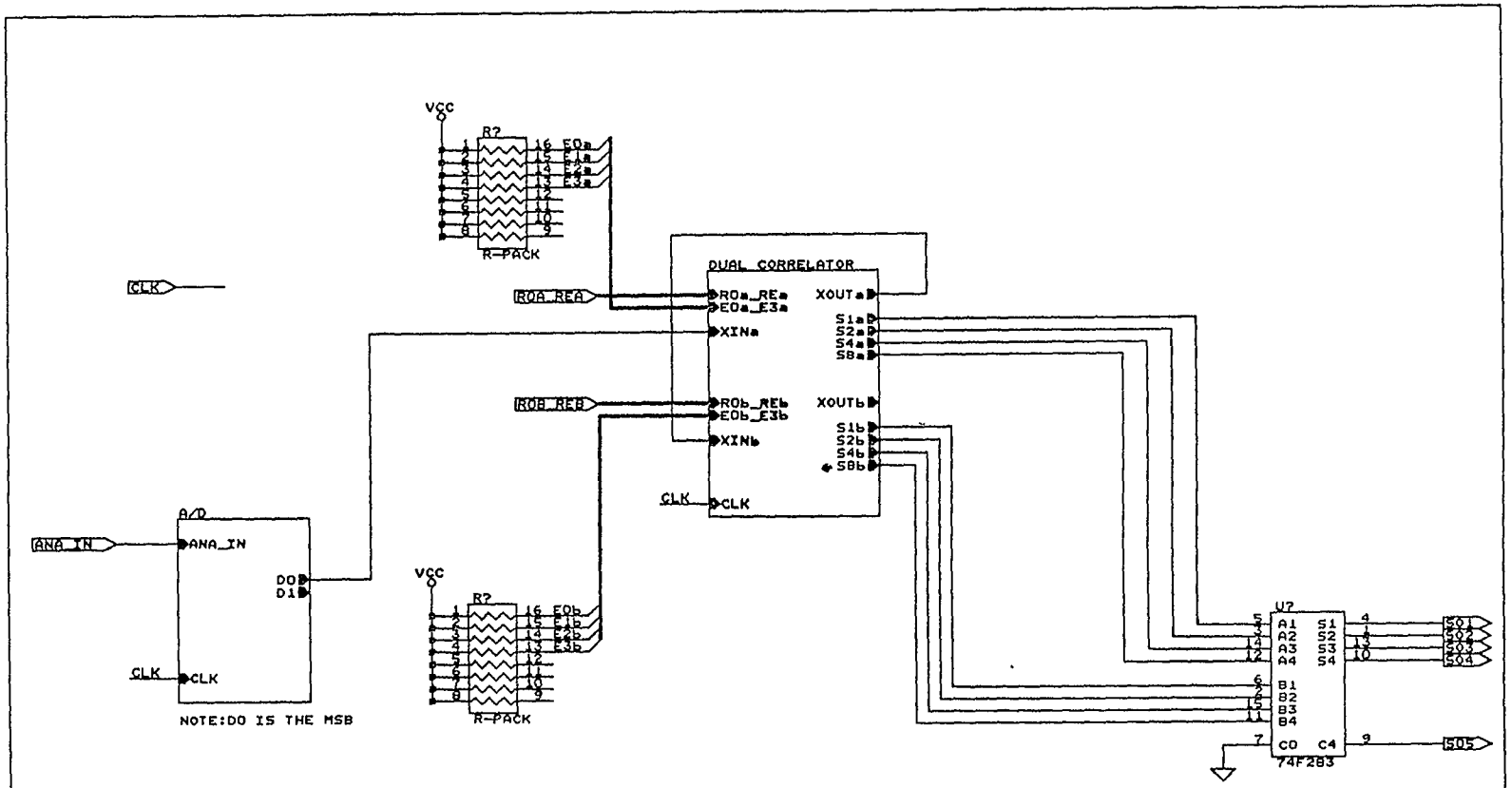


Figure 3.17 30 X 1 CORRELATOR

R. CHADWICK NJIT		
Title: 30X1 CORRELATOR		
Size: B	Document Number	REV: 0.1
Date: November 28, 1990	Sheet of	

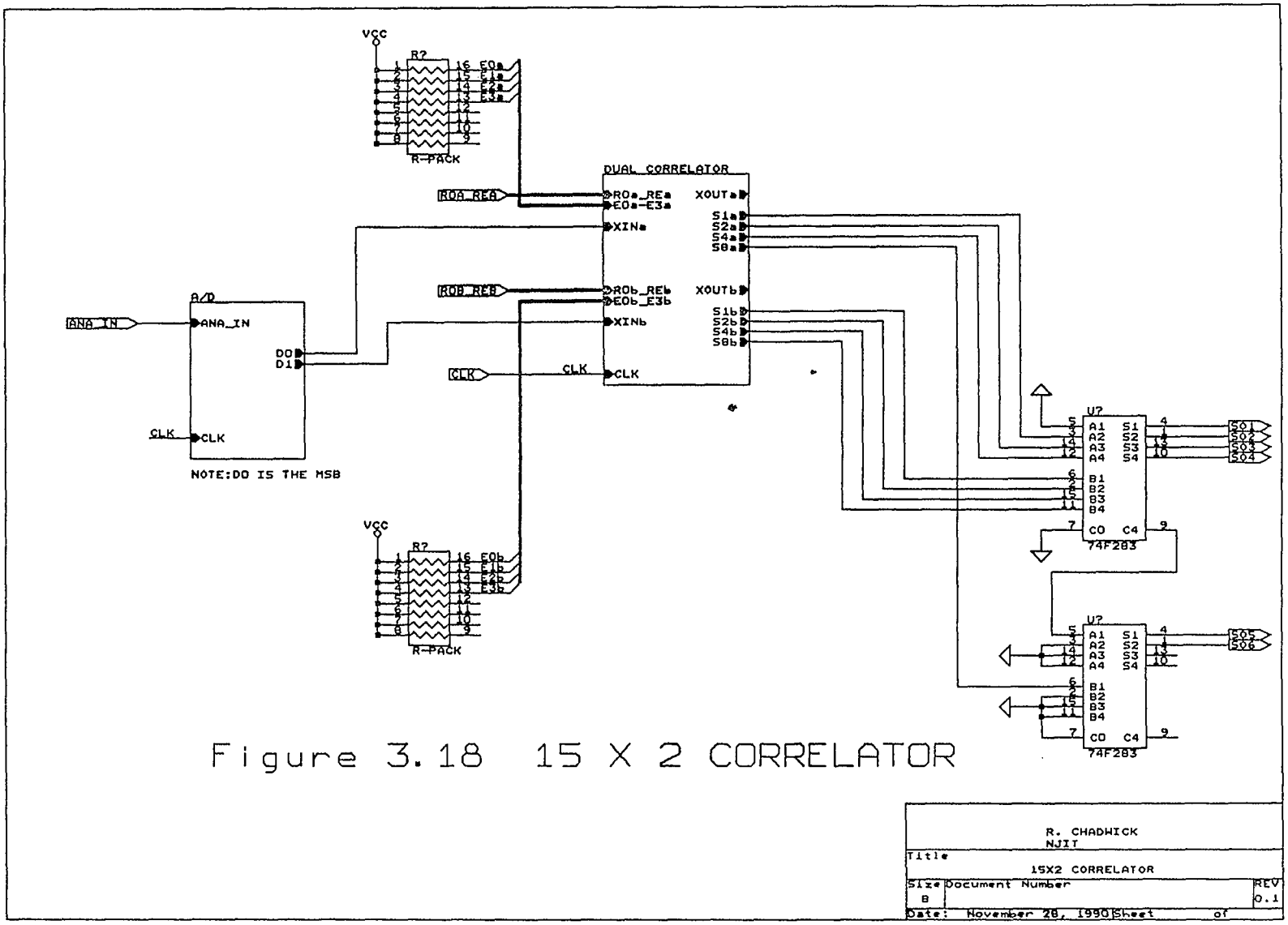


Figure 3.18 15 X 2 CORRELATOR

divided by two, or simply shifted one place. If more bits from the A/D were used, each bit is divided by two more than the preceding. The results are then added together to form the final correlation value.

As shown in figure 3.19, correlators may be cascaded in parallel and series to increase the correlation dimensions, n X m. The partial sums from the correlators in series are first added, scaled, and then the scaled sums are added together.

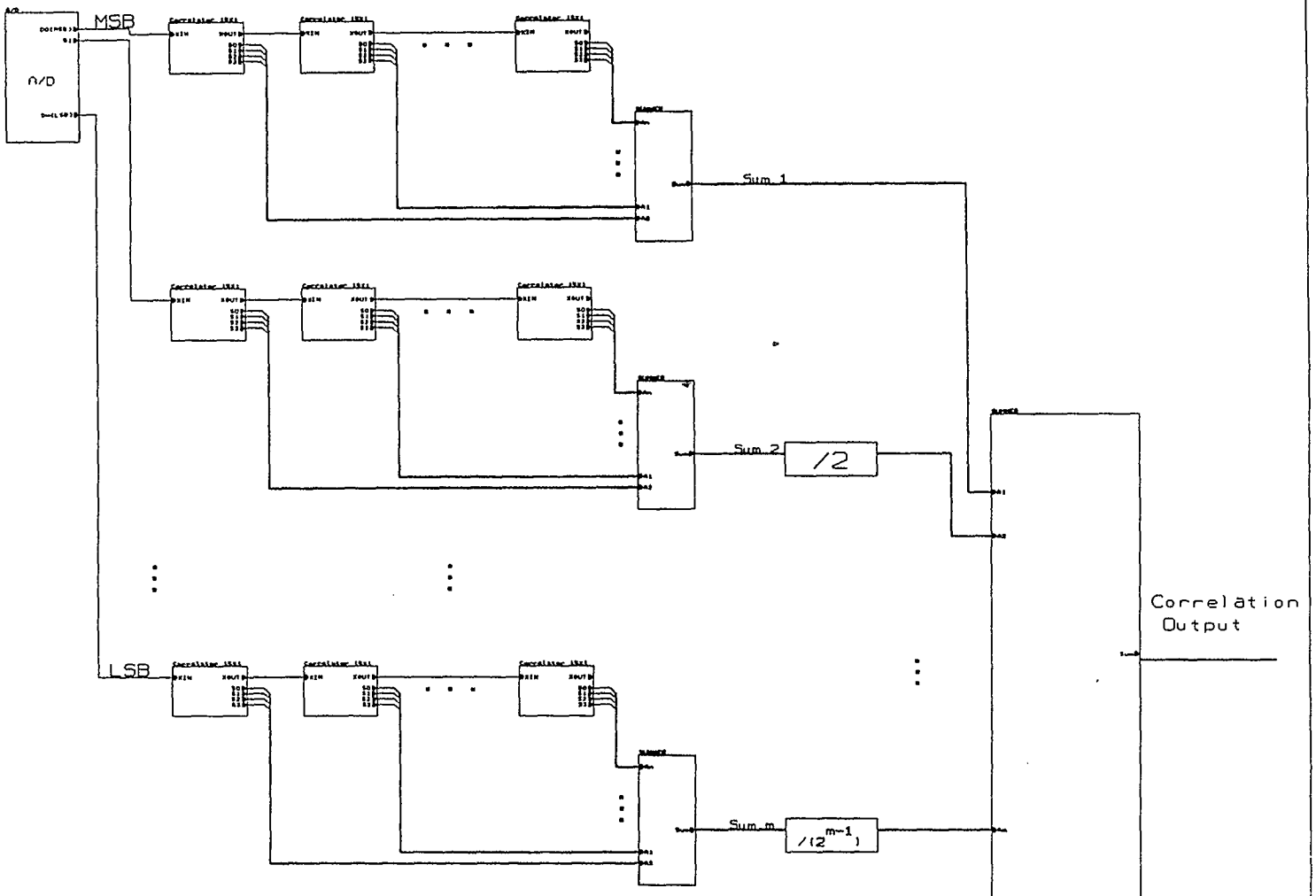


Figure 3.19 n X m Correlator Block Diagram

CHAPTER 4
SIMULATION and CONCLUSION

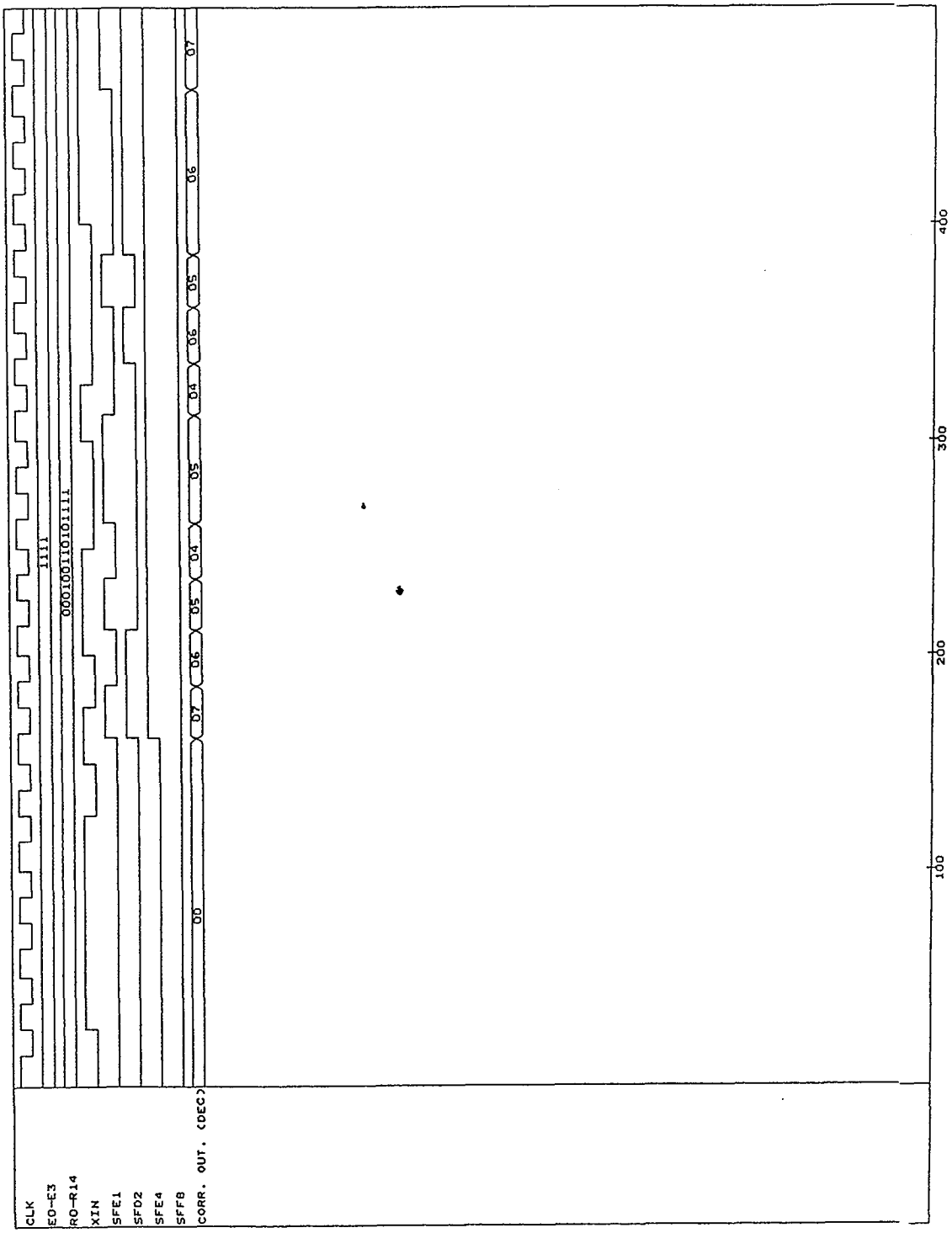
4.1 Simulation

The final placed and routed Xilinx correlator design was translated to a file compatible with Orcad's VST simulator. A stimulus was created using Orcad's stimulus editor. Correlator a's enable bits were all set high, setting the correlator to its maximum length of 15 bits. The reference code bits, R0 - RE, were set to 0001 0011 0101 111, a typical linear maximal sequence that is often used in spread spectrum communications. The same code sequence was also serially shifted into the Xin input of the correlator. The design was simulated at its predicted maximum internal operating frequency of 40 Mhz. The simulation appears in figure 4.1; the time scale is in nanoseconds. The output of the correlator, after the initial conditions were shifted out of the correlator, was a thumb-tack response; the output of the correlator was seven until a correlation, and then the output jumped to 15 decimal. At t=1000ns, the reference code bits, R0 - RE, were all toggled. The output of the correlator, after the initial conditions are shifted out, was a thumb-tack response. The uncorrelated value was eight, and the correlated value was 0. At t=1300ns, the correlator's enable bits were all toggled, and set the

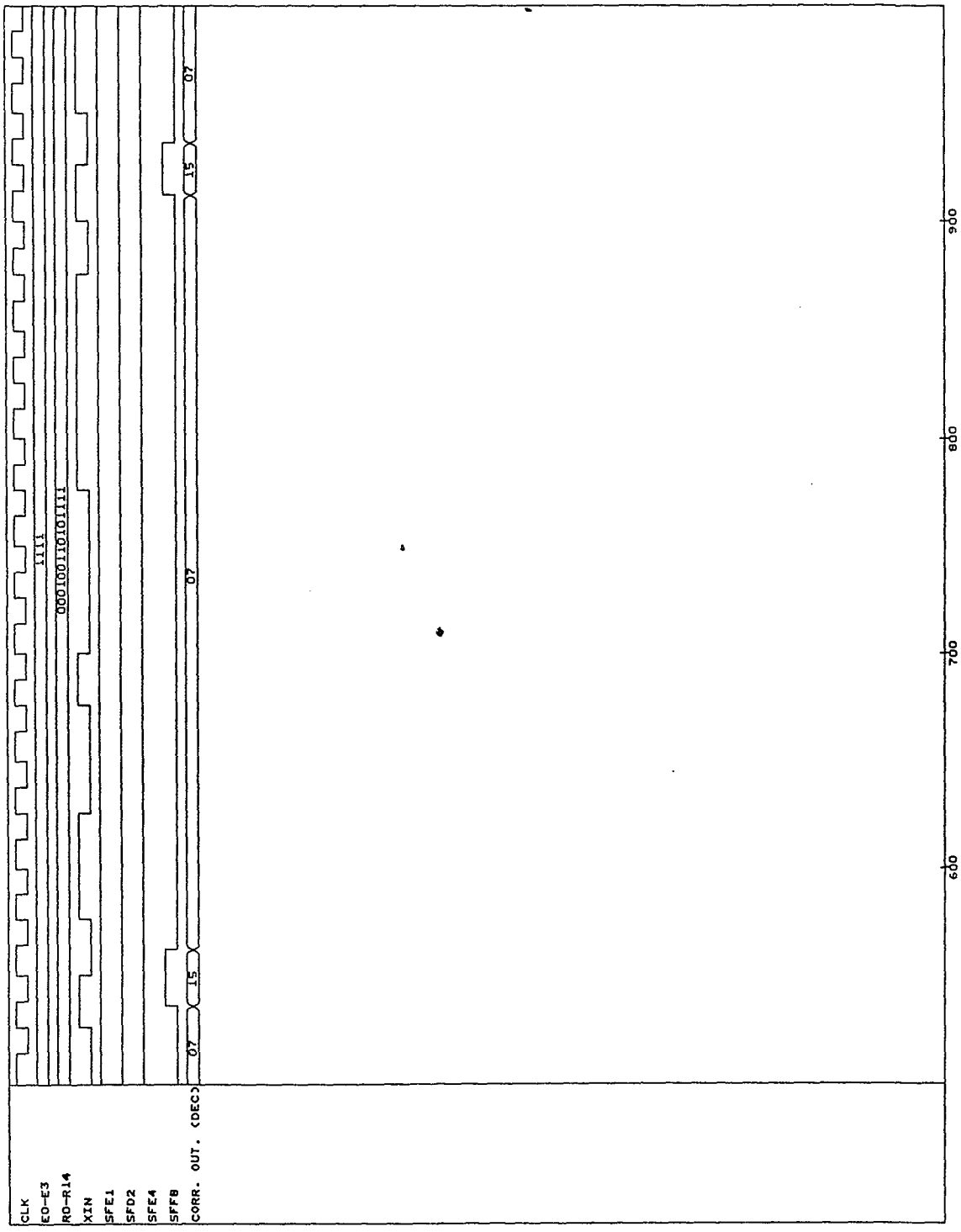
correlator's length to seven stages. The correlator's active reference code bits, R1, R3, R5, R7, R9, R11, and R13, were set to 0010 111, a seven bit linear maximal sequence. The same sequence was also shifted into the Xin input; again, a thumb-tack response was recorded. The uncorrelated value was three, and the correlated value was seven. The simulated results were identical to the expected results.

The simulation in figure 4.2 is identical to the simulation in figure 4.1, except that all the states of the nets associated with correlator a are displayed. The nets associated with the matching sections (net names begin with an M) and the shift-register (net names begin with an X) were displayed as binary buses. For example, M0, M1 and M2 were displayed as a binary bus. M0 is the most significant bit, the left bit, and M2 is the least significant bit. The output of each adder (nets that begin with an S) was displayed as a decimal bus. The most significant bit out of the adder is the most significant bit on the bus displayed. This simulation was checked, and every net in the correlator was toggled at least once.

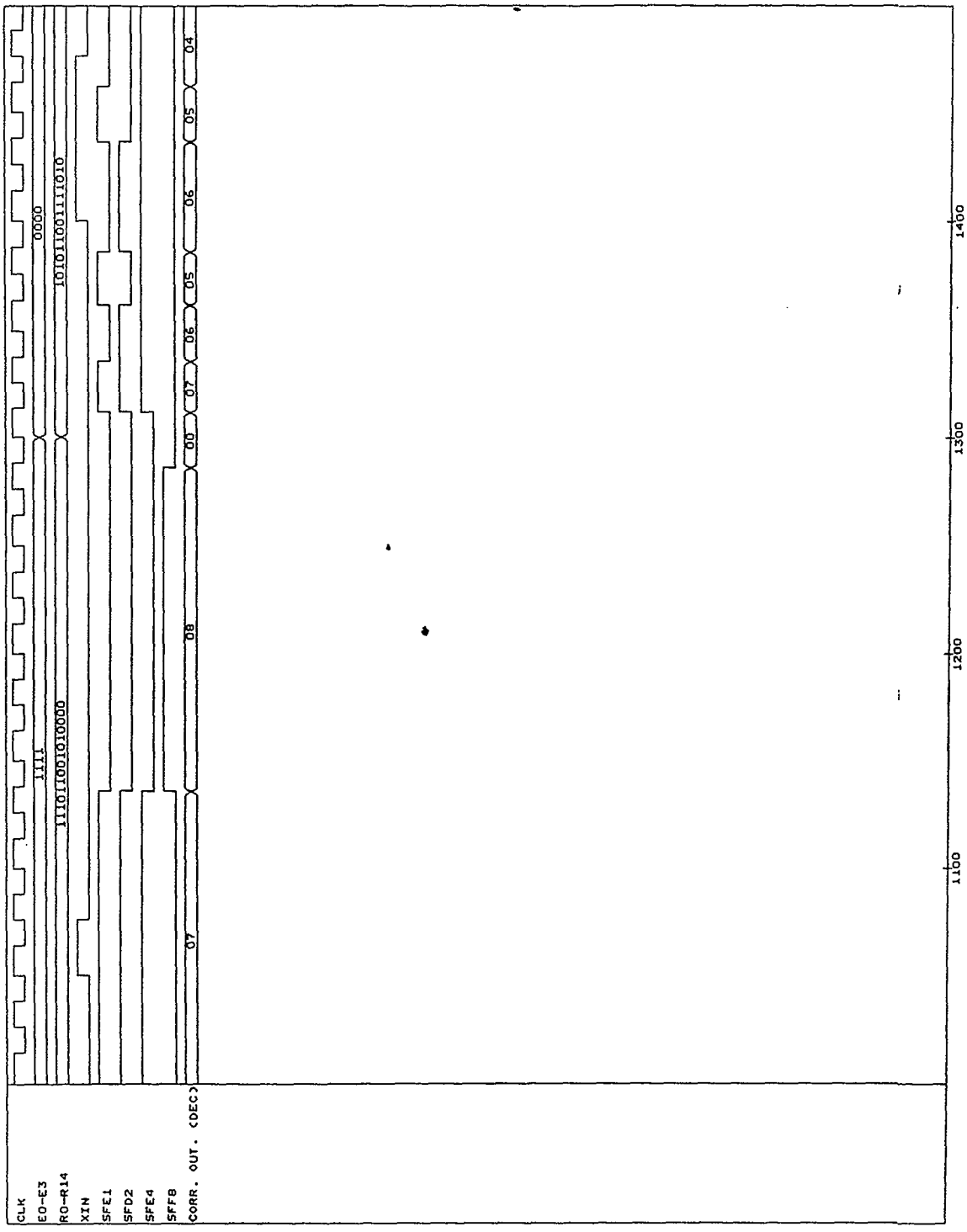
The design was verified for functionality and at an operating frequency of 40 Mhz. Correlator b was simulated in an identical manner.



Time scale is in nanoseconds.
Figure 4.1. Simulation of Correlator a.

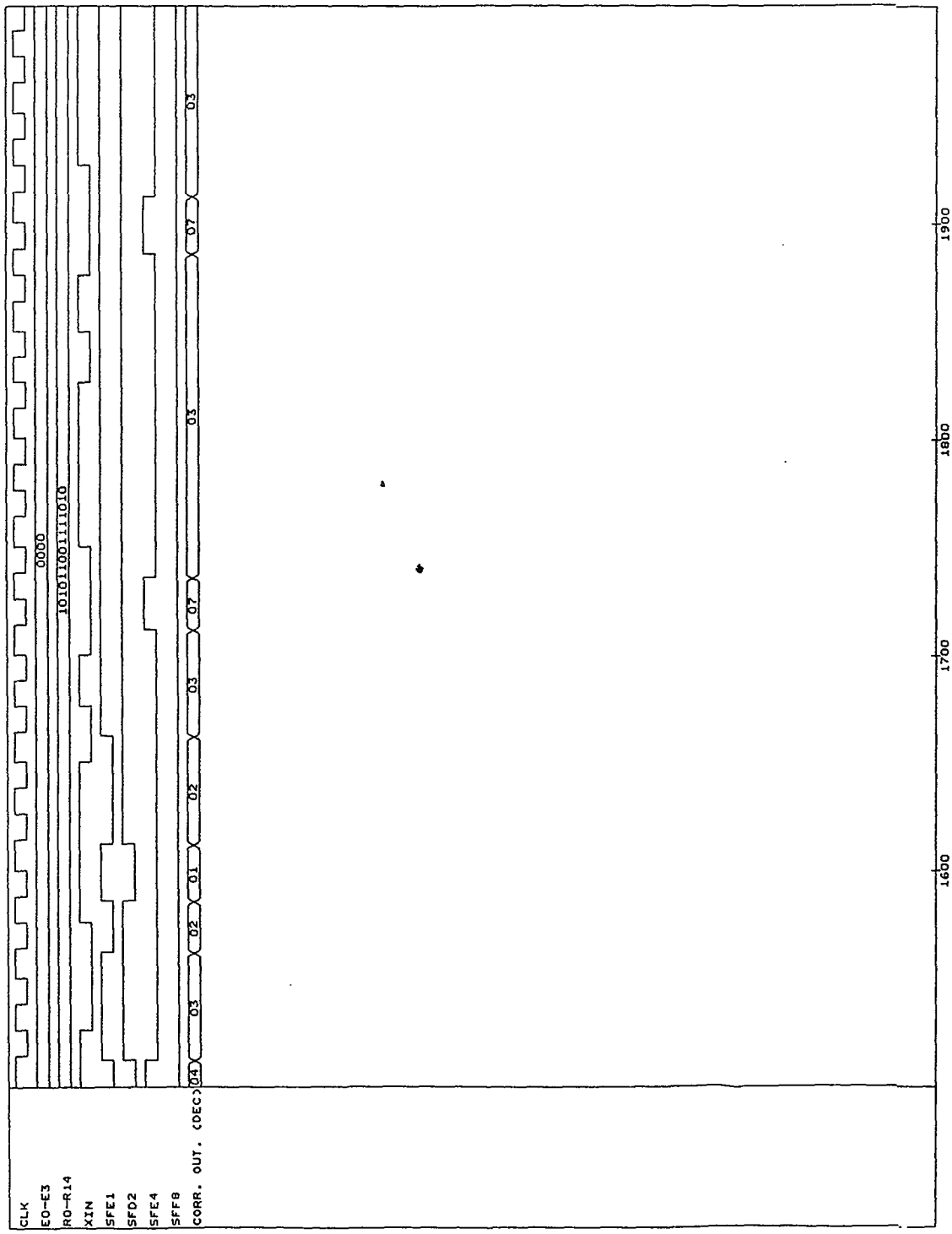


Time scale is in nanoseconds.
Figure 4.1 (continued). Simulation of correlator a.

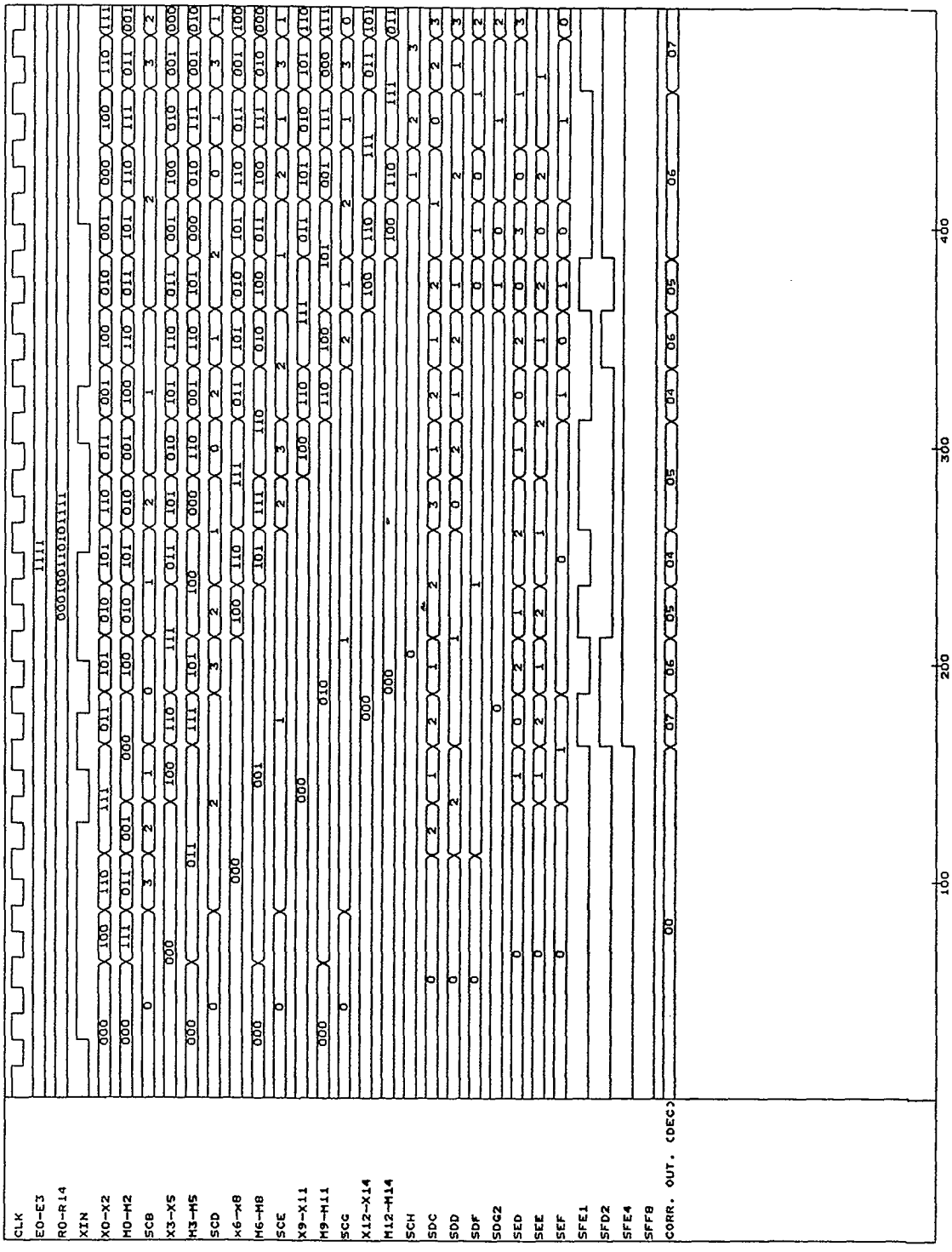


Time scale is in nanoseconds.

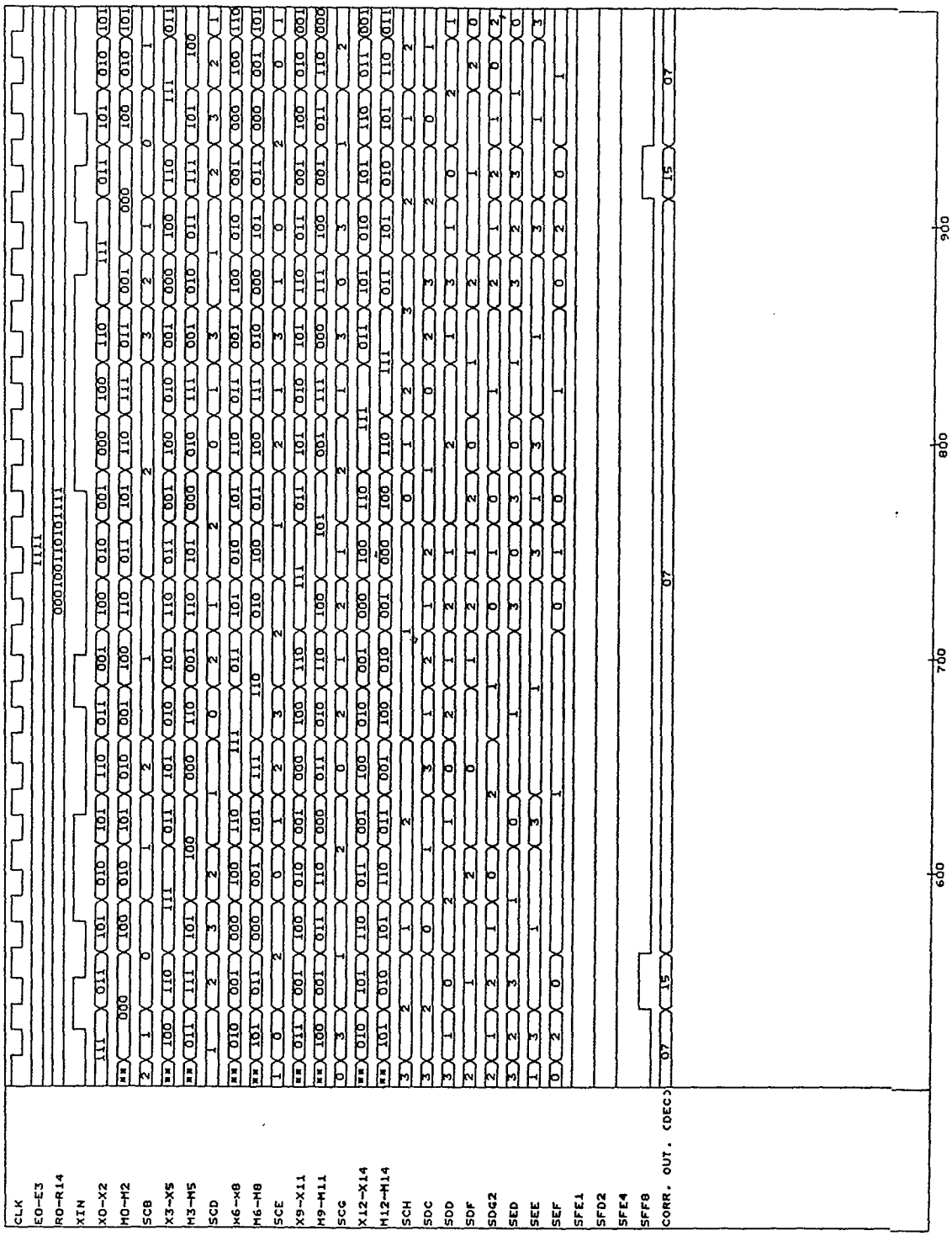
Figure 4.1 (continued). Simulation of correlator a.



Time scale is in nanoseconds.
Figure 4.1(continued). Simulation of correlator a.

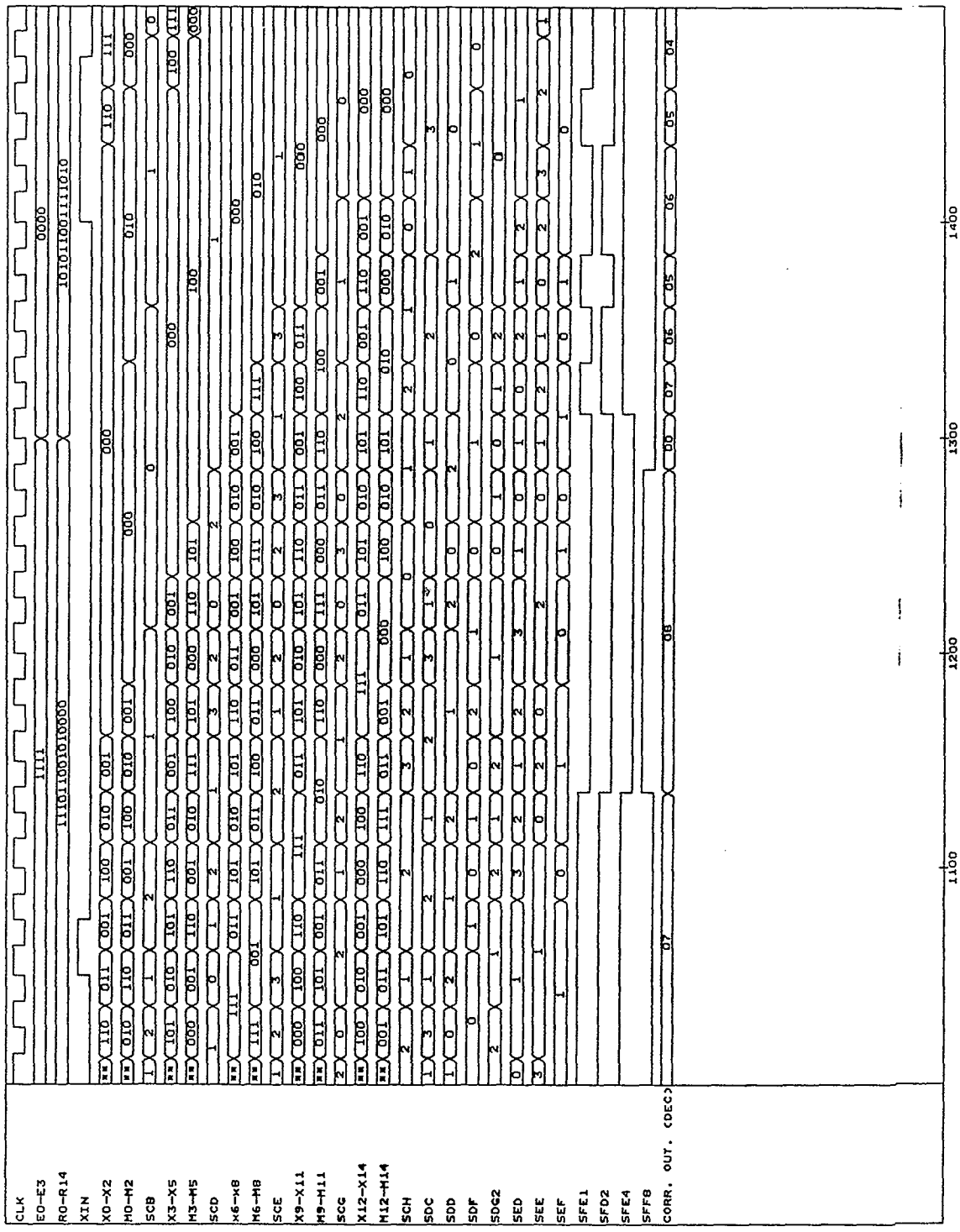


Time scale is in nanoseconds.
Figure 4.2. Simulation of correlator a.



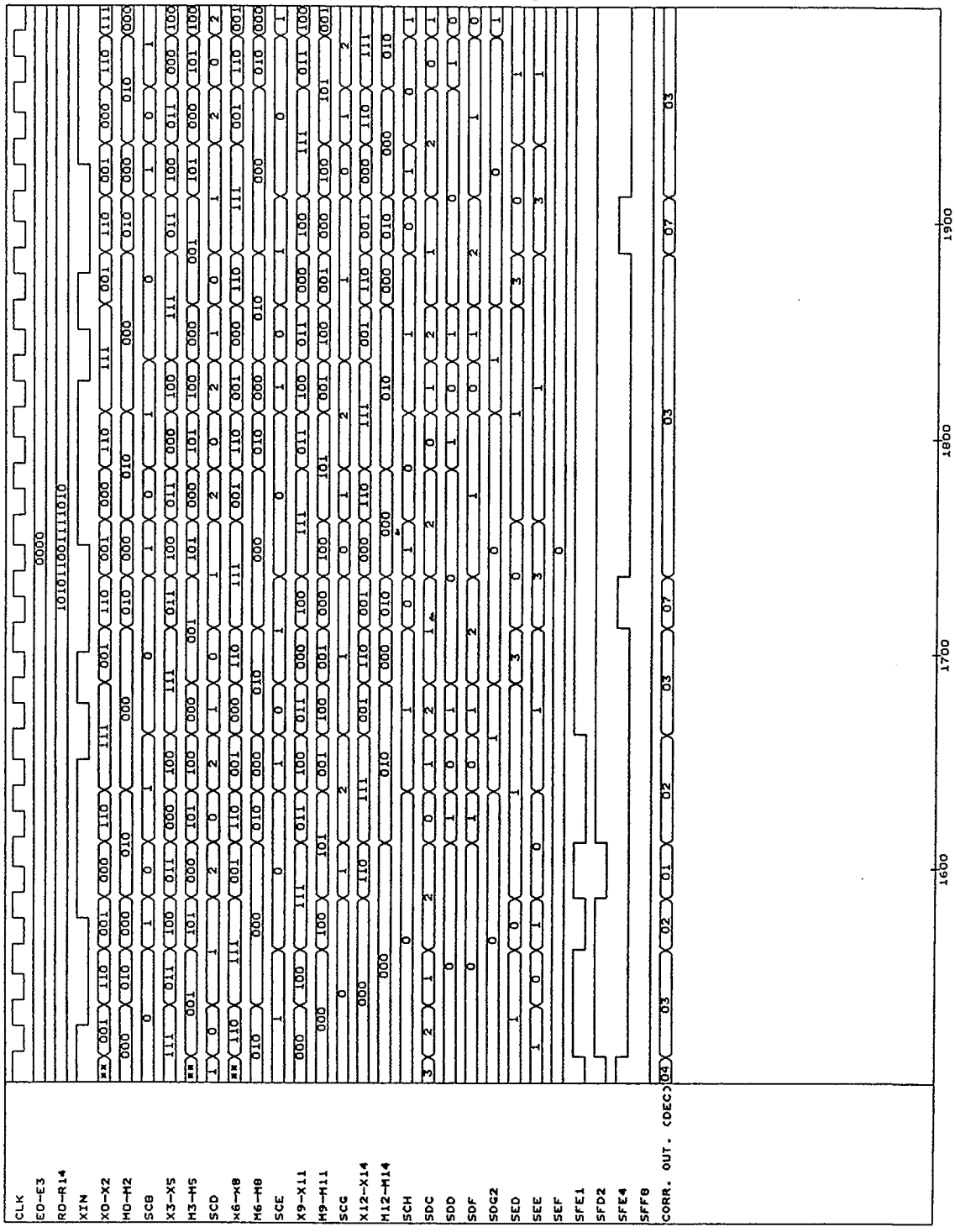
Time scale is in nanoseconds.

Figure 4.2 (continued). Simulation of correlator a.



Time scale is in nanoseconds.

Figure 4.2 (continued). Simulation of correlator a.



Time scale is in nanoseconds.

Figure 4.2(continued). Simulation of correlator a.

4.2 Conclusion

The design and application, in spread spectrum communications, of a digital correlator implemented in a Xilinx Field Programmable Gate Array has been presented. The correlator design was implemented in a Xilinx 3020 FPGA, and consists of two independent correlators; the length n of each correlator was designed to be variable from seven to 15 bits. Each correlator's dimension m, the number of bits that the input signal is quantized to, was designed to be one. The correlation dimensions may be expanded by cascading correlators in series and parallel. The dimensions may be varied from a minimum of 7 X 1 to any practical maximum. The limiting factors are power consumption, cost, size and the ability of an external adder to sum the results from all of the correlators.

The Xilinx FPGA offered many of the advantages of a CMOS Application Specific Integrated Circuit, ASIC, without the high Non-Recurring Engineering costs. The FPGA is a low power device, and contains two complete correlators on one chip.

The Correlator design was estimated and simulated to be useful up to a maximum frequency of 40 Mhz. The maximum system operating frequency, using Gazelle's GA22V10 Programmable Logic Device as a register to buffer the input and output of the FPGA, was estimated to be 40 Mhz. This is a significant speed improvement over commercially available devices such as TRW's 2023 Correlator which has a maximum operating frequency of 30 Mhz.

APPENDIX
A1 Dual Correlator Net Delays

Querynet: RAY05A5.LCA (3020PC68-70), XACT 2.12, Sat Nov 24 15:04:56 1990

---- _aE0	P12.Q (E0a). . .	25	DA.B (X01_a)
		~29	EB.D (M01_a)
		~28	EA.B (X23_a)
		~27	DC.B (M23_a)
		22	DB.A (X45_a)
		19	BC.E (M45_a)
		12	CA.E (X67_a)
		18	CB.A (M67_a)
---- _aE1	P8.Q (E1a) . . .	6	BA.A (X89_a)
		6	BB.C (M89_a)
		4	AA.B (XAB_a)
		5	AB.E (MAB_a)
---- _aE2	P2.Q (E2a) . . .	6	AC.A (XCD_a)
		5	AD.A (MCD_a)
---- _aE3	P65.Q (E3a). . .	3	AG.C (ME_a)
---- _aMA	AB.X (MAB_a) . .	3	BD.C (SCG_a)
---- _aMB	AB.Y (MAB_a) . .	5	BD.B (SCG_a)
---- _aMC	AD.X (MCD_a) . .	1	AE.B (SCH_a)
---- _aMD	AD.Y (MCD_a) . .	2	AE.C (SCH_a)
---- _aME	AG.X (ME_a). . .	6	AE.A (SCH_a)
---- _aM0	EB.Y (M01_a) . .	3	DD.E (SCB_a)
---- _aM1	EB.X (M01_a) . .	5	DD.A (SCB_a)
---- _aM2	DC.X (M23_a) . .	1	DD.B (SCB_a)
---- _aM3	DC.Y (M23_a) . .	3	CD.D (SCD_a)
---- _aM4	BC.X (M45_a) . .	2	CD.A (SCD_a)
---- _aM5	BC.Y (M45_a) . .	3	CD.B (SCD_a)
---- _aM6	CB.X (M67_a) . .	1	CC.B (SCE_a)
---- _aM7	CB.Y (M67_a) . .	2	CC.C (SCE_a)
---- _aM8	BB.X (M89_a) . .	3	CC.A (SCE_a)
---- _aM9	BB.Y (M89_a) . .	3	BD.A (SCG_a)
---- _aRA	P9.Q (RAa) . . .	3	AB.A (MAB_a)
---- _aRB	P7.Q (RBa) . . .	3	AB.D (MAB_a)
---- _aRC	P66.Q (RCa). . .	6	AD.C (MCD_a)
---- _aRD	P68.Q (RDa). . .	4	AD.E (MCD_a)
---- _aRE	P64.Q (REa). . .	2	AG.A (ME_a)
---- _aR0	P20.Q (R0a). . .	4	EB.B (M01_a)
---- _aR1	P19.Q (R1a). . .	3	EB.E (M01_a)
---- _aR2	P5.Q (R2a) . . .	8	DC.A (M23_a)

---- _aR3	P17.Q (R3a). . .	5	DC.D (M23_a)
---- _aR4	P67.Q (R4a). . .	7	BC.A (M45_a)
---- _aR5	P4.Q (R5a) . . .	3	BC.B (M45_a)
---- _aR6	P16.Q (R6a). . .	3	CB.D (M67_a)
---- _aR7	P14.Q (R7a). . .	3	CB.E (M67_a)
---- _aR8	P13.Q (R8a). . .	4	BB.D (M89_a)
---- _aR9	P6.Q (R9a) . . .	3	BB.A (M89_a)
---- _aSCB1	DD.X (SCB_a) . .	1	DE.B (SDC_a)
---- _aSCB2	DD.Y (SCB_a) . .	3	CE.D (SDD_a)
---- _aSCD1	CD.Y (SCD_a) . .	3	DE.A (SDC_a)
---- _aSCD2	CD.X (SCD_a) . .	1	CE.B (SDD_a)
---- _aSCE1	CC.Y (SCE_a) . .	5	DE.C (SDC_a)
---- _aSCE2	CC.X (SCE_a) . .	3	CE.C (SDD_a)
---- _aSCG1	BD.Y (SCG_a) . .	5	CF.D (SDF_a)
---- _aSCG2	BD.X (SCG_a) . .	1	BE.B (SDG_a)
---- _aSCH1	AE.X (SCH_a) . .	4	CF.B (SDF_a)
---- _aSCH2	AE.Y (SCH_a) . .	1	BE.A (SDG_a)
---- _aSDC1	DE.Y (SDC_a) . .	3	CG.E (SEE_a)
---- _aSDC2	DE.X (SDC_a) . .	5	BF.A (SED_a)
---- _aSDD2	CE.Y (SDD_a) . .	3	BF.D (SED_a)
---- _aSDD4	CE.X (SDD_a) . .	5	BG.C (SEF_a)
---- _aSDF1	CF.X (SDF_a) . .	1	CG.B (SEE_a)
---- _aSDF2	CF.Y (SDF_a) . .	2	CG.C (SEE_a)
---- _aSDG2	BE.X (SDG_a) . .	1	BF.B (SED_a)
---- _aSDG4	BE.Y (SDG_a) . .	3	BG.E (SEF_a)
---- _aSED2	BF.Y (SED_a) . .	6	BH.A (SFF_a)
		7	CH.C (SFD_a)
		8	AH.A (SFE_a)
---- _aSED4	BF.X (SED_a) . .	5	BH.E (SFF_a)
		8	AH.C (SFE_a)
---- _aSEE1	CG.Y (SEE_a) . .	5	AH.E (SFE_a)
---- _aSEE2	CG.X (SEE_a) . .	4	BH.D (SFF_a)
		1	CH.B (SFD_a)
		5	AH.B (SFE_a)
---- _aSEF4	BG.X (SEF_a) . .	1	BH.B (SFF_a)
		4	AH.D (SFE_a)
---- _aSEF8	BG.Y (SEF_a) . .	2	BH.C (SFF_a)
---- _aSFD2	CH.Y (SFD_a) . .	3	P56.O (S2a)
---- _aSFE1	AH.X (SFE_a) . .	4	P62.O (S1a)
---- _aSFE4	AH.Y (SFE_a) . .	5	P63.O (S4a)
---- _aSFF8	BH.X (SFF_a) . .	4	P61.O (S8a)
---- _aXA	AA.X (XAB_a) . .	1	AB.B (MAB_a)

---- _aXB	AA.Y (XAB_a) . .	5	AC.D (XCD_a)
		3	AB.C (MAB_a)
---- _aXC	AC.Y (XCD_a) . .	3	AD.D (MCD_a)
		4	P3.O (XOUTa)
---- _aXD	AC.X (XCD_a) . .	5	AF.C (XE_a)
		1	AD.B (MCD_a)
---- _aXE	AF.X (XE_a). . .	1	AG.B (ME_a)
---- _aXIN.	P15.Q (XINa) . .	3	DA.A (X01_a)
---- _aX0	DA.X (X01_a) . .	3	EB.A (M01_a)
---- _aX1	DA.Y (X01_a) . .	1	EA.A (X23_a)
		3	EB.C (M01_a)
---- _aX2	EA.X (X23_a) . .	4	DC.E (M23_a)
---- _aX3	EA.Y (X23_a) . .	3	DB.D (X45_a)
		3	DC.C (M23_a)
---- _aX4	DB.Y (X45_a) . .	5	BC.C (M45_a)
---- _aX5	DB.X (X45_a) . .	8	CA.A (X67_a)
		7	BC.D (M45_a)
---- _aX6	CA.X (X67_a) . .	1	CB.B (M67_a)
---- _aX7	CA.Y (X67_a) . .	1	BA.D (X89_a)
		3	CB.C (M67_a)
---- _aX8	BA.X (X89_a) . .	1	BB.B (M89_a)
---- _aX9	BA.Y (X89_a) . .	1	AA.D (XAB_a)
		3	BB.E (M89_a)
---- _bE0	P33.Q (E0b). . .	25	GA.B (X01_b)
		~30	GC.E (M01_b)
		27	FA.C (X23_b)
		18	FB.D (M23_b)
		19	HA.D (X45_b)
		16	GB.C (M45_b)
		15	HB.D (X67_b)
		11	HC.C (M67_b)
---- _bE1	P41.Q (E1b). . .	9	HD.A (X89_b)
		9	HE.A (M89_b)
		9	GE.C (XAB_b)
		8	HF.B (MAB_b)
---- _bE2	P53.Q (E2b). . .	6	DF.A (XCD_b)
		5	DH.B (MCD_b)
---- _bE3	P49.Q (E3b). . .	3	GH.A (ME_b)
---- _bMA	HF.Y (MAB_b) . .	3	GG.D (SCG_b)
---- _bMB	HF.X (MAB_b) . .	3	GG.A (SCG_b)
---- _bMC	DH.Y (MCD_b) . .	1	EH.A (SCH_b)
---- _bMD	DH.X (MCD_b) . .	4	EH.B (SCH_b)

----	_bME	GH.Y (ME_b) . . .	4	EH.C	(SCH_b)
----	_bM0	GC.Y (M01_b) . .	3	FD.B	(SCB_b)
----	_bM1	GC.X (M01_b) . .	3	FD.E	(SCB_b)
----	_bM2	FB.Y (M23_b) . .	4	FD.A	(SCB_b)
----	_bM3	FB.X (M23_b) . .	1	FC.B	(SCD_b)
----	_bM4	GB.Y (M45_b) . .	3	FC.D	(SCD_b)
----	_bM5	GB.X (M45_b) . .	4	FC.C	(SCD_b)
----	_bM6	HC.Y (M67_b) . .	3	GD.D	(SCE_b)
----	_bM7	HC.X (M67_b) . .	3	GD.A	(SCE_b)
----	_bM8	HE.X (M89_b) . .	4	GD.C	(SCE_b)
----	_bM9	HE.Y (M89_b) . .	3	GG.B	(SCG_b)
----	_bRA	P38.Q (RAb) . . .	3	HF.E	(MAB_b)
----	_bRB	P47.Q (RBb) . . .	6	HF.C	(MAB_b)
----	_bRC	P51.Q (RCb) . . .	4	DH.D	(MCD_b)
----	_bRD	P55.Q (RDb) . . .	3	DH.E	(MCD_b)
----	_bRE	P46.Q (REb) . . .	3	GH.C	(ME_b)
----	_bR0	P32.Q (R0b) . . .	6	GC.B	(M01_b)
----	_bR1	P31.Q (R1b) . . .	4	GC.D	(M01_b)
----	_bR2	P23.Q (R2b) . . .	7	FB.C	(M23_b)
----	_bR3	P21.Q (R3b) . . .	3	FB.E	(M23_b)
----	_bR4	P22.Q (R4b) . . .	3	GB.A	(M45_b)
----	_bR5	P29.Q (R5b) . . .	4	GB.E	(M45_b)
----	_bR6	P36.Q (R6b) . . .	5	HC.A	(M67_b)
--C-	_bR7	P40.Q (R7b) . . .	9	HC.D	(M67_b)
----	_bR8	P39.Q (R8b) . . .	4	HE.E	(M89_b)
----	_bR9	P37.Q (R9b) . . .	3	HE.D	(M89_b)
----	_bSCB1	FD.X (SCB_b) . .	1	FE.B	(SDC_b)
----	_bSCB2	FD.Y (SCB_b) . .	3	EE.D	(SDD_b)
----	_bSCD1	FC.Y (SCD_b) . .	3	FE.E	(SDC_b)
----	_bSCD2	FC.X (SCD_b) . .	5	EE.A	(SDD_b)
----	_bSCE1	GD.X (SCE_b) . .	3	FE.C	(SDC_b)
----	_bSCE2	GD.Y (SCE_b) . .	4	EE.C	(SDD_b)
----	_bSCG1	GG.X (SCG_b) . .	3	HH.B	(SDF_b)
----	_bSCG2	GG.Y (SCG_b) . .	3	FH.D	(SDG_b)
----	_bSCH1	EH.X (SCH_b) . .	7	HH.A	(SDF_b)
----	_bSCH2	EH.Y (SCH_b) . .	1	FH.A	(SDG_b)
----	_bSDC1	FE.Y (SDC_b) . .	3	GF.A	(SEE_b)
----	_bSDC2	FE.X (SDC_b) . .	1	FF.B	(SED_b)
----	_bSDD2	EE.Y (SDD_b) . .	4	FF.A	(SED_b)
----	_bSDD4	EE.X (SDD_b) . .	1	EF.B	(SEF_b)
----	_bSDF1	HH.Y (SDF_b) . .	7	GF.C	(SEE_b)
----	_bSDF2	HH.X (SDF_b) . .	8	GF.B	(SEE_b)

---- _bSDG2	FH.Y (SDG_b) . .	7	FF.C (SED_b)
---- _bSDG4	FH.X (SDG_b) . .	7	EF.D (SEF_b)
---- _bSED2	FF.X (SED_b) . .	3	EG.C (SFF_b)
		7	HG.E (SFD_b)
		1	FG.B (SFE_b)
---- _bSED4	FF.Y (SED_b) . .	3	EG.D (SFF_b)
		3	FG.A (SFE_b)
---- _bSEE1	GF.Y (SEE_b) . .	3	FG.D (SFE_b)
---- _bSEE2	GF.X (SEE_b) . .	8	EG.A (SFF_b)
		3	HG.A (SFD_b)
		5	FG.E (SFE_b)
---- _bSEF4	EF.X (SEF_b) . .	1	EG.B (SFF_b)
		4	FG.C (SFE_b)
---- _bSEF8	EF.Y (SEF_b) . .	2	EG.E (SFF_b)
---- _bSFD2	HG.Y (SFD_b) . .	2	P42.O (S2b)
---- _bSFE1	FG.Y (SFE_b) . .	7	P43.O (S1b)
---- _bSFE4	FG.X (SFE_b) . .	3	P48.O (S4b)
---- _bSFF8	EG.X (SFF_b) . .	2	P50.O (S8b)
---- _bXA	GE.X (XAB_b) . .	2	HF.A (MAB_b)
---- _bXB	GE.Y (XAB_b) . .	8	DF.D (XCD_b)
		5	HF.D (MAB_b)
---- _bXC	DF.Y (XCD_b) . .	4	DH.A (MCD_b)
		4	P54.O (XOUTb)
---- _bXD	DF.X (XCD_b) . .	1	DG.B (XE_b)
		3	DH.C (MCD_b)
---- _bXE	DG.Y (XE_b). . .	6	GH.B (ME_b)
---- _bXIN.	P24.Q (XINb) . .	3	GA.A (X01_b)
---- _bX0	GA.X (X01_b) . .	3	GC.C (M01_b)
---- _bX1	GA.Y (X01_b) . .	1	FA.D (X23_b)
		3	GC.A (M01_b)
---- _bX2	FA.X (X23_b) . .	1	FB.B (M23_b)
---- _bX3	FA.Y (X23_b) . .	6	HA.A (X45_b)
		2	FB.A (M23_b)
---- _bX4	HA.Y (X45_b) . .	3	GB.D (M45_b)
---- _bX5	HA.X (X45_b) . .	1	HB.B (X67_b)
		3	GB.B (M45_b)
---- _bX6	HB.Y (X67_b) . .	2	HC.E (M67_b)
---- _bX7	HB.X (X67_b) . .	3	HD.C (X89_b)
		1	HC.B (M67_b)
---- _bX8	HD.X (X89_b) . .	1	HE.B (M89_b)
---- _bX9	HD.Y (X89_b) . .	4	GE.D (XAB_b)
		3	HE.C (M89_b)

---- CLK.	GCLK.O	4	P67.IK (R4a)
		4	P4.IK (R5a)
		3	P16.IK (R6a)
		3	P20.IK (R0a)
		3	P19.IK (R1a)
		4	P5.IK (R2a)
		3	P17.IK (R3a)
		3	P15.IK (XINa)
		3	DE.K (SDC_a)
		3	CF.K (SDF_a)
		3	BE.K (SDG_a)
		3	AE.K (SCH_a)
		3	BD.K (SCG_a)
		3	CC.K (SCE_a)
		3	CD.K (SCD_a)
		3	DD.K (SCB_a)
		3	EB.K (M01_a)
		3	DC.K (M23_a)
		3	BC.K (M45_a)
		3	CB.K (M67_a)
		3	BB.K (M89_a)
		3	AD.K (MCD_a)
		3	AG.K (ME_a)
		3	AF.K (XE_a)
		3	AC.K (XCD_a)
		3	AA.K (XAB_a)
		3	BA.K (X89_a)
		3	CA.K (X67_a)
		3	DB.K (X45_a)
		3	EA.K (X23_a)
		3	DA.K (X01_a)
		3	CE.K (SDD_a)
		3	AB.K (MAB_a)
		4	P64.IK (REa)
		4	P68.IK (RDa)
		4	P66.IK (RCa)
		4	P65.IK (E3a)
		4	P7.IK (RBa)
		4	P9.IK (RAa)
		4	P6.IK (R9a)
		4	P8.IK (E1a)
		3	P13.IK (R8a)

```
3 P14.IK (R7a)
4 P3.OK (XOUTa)
3 BF.K (SED_a)
3 CG.K (SEE_a)
3 BH.K (SFF_a)
3 CH.K (SFD_a)
3 AH.K (SFE_a)
3 BG.K (SEF_a)
4 P62.OK (S1a)
3 P56.OK (S2a)
3 P31.IK (R1b)
3 P23.IK (R2b)
3 P21.IK (R3b)
3 P24.IK (XINb)
3 DG.K (XE_b)
3 DF.K (XCD_b)
3 GE.K (XAB_b)
3 HD.K (X89_b)
3 HB.K (X67_b)
3 HA.K (X45_b)
3 FA.K (X23_b)
3 GA.K (X01_b)
3 GC.K (M01_b)
3 FB.K (M23_b)
3 GB.K (M45_b)
3 HC.K (M67_b)
3 HE.K (M89_b)
3 DH.K (MCD_b)
3 GH.K (ME_b)
3 HF.K (MAB_b)
3 EH.K (SCH_b)
3 GG.K (SCG_b)
3 GD.K (SCE_b)
3 FC.K (SCD_b)
3 FD.K (SCB_b)
3 FE.K (SDC_b)
3 HH.K (SDF_b)
3 FH.K (SDG_b)
3 EE.K (SDD_b)
3 FF.K (SED_b)
3 GF.K (SEE_b)
3 EF.K (SEF_b)
```

```
                                                    3 EG.K (SFF_b)
                                                    3 HG.K (SFD_b)
                                                    3 FG.K (SFE_b)
                                                    3 P50.OK (S8b)
                                                    3 P48.OK (S4b)
                                                    3 P42.OK (S2b)
                                                    3 P43.OK (S1b)
                                                    3 P22.IK (R4b)
                                                    3 P29.IK (R5b)
                                                    3 P36.IK (R6b)
                                                    3 P39.IK (R8b)
                                                    3 P37.IK (R9b)
                                                    3 P38.IK (RAb)
                                                    3 P53.IK (E2b)
                                                    3 P49.IK (E3b)
                                                    3 P51.IK (RCb)
                                                    3 P46.IK (REb)
                                                  3 P54.OK (XOUTb)
                                                    3 P33.IK (E0b)
                                                    3 P12.IK (E0a)
                                                    3 P32.IK (R0b)
                                                    3 P41.IK (E1b)
                                                    4 P2.IK (E2a)
                                                    4 P61.OK (S8a)
                                                    4 P63.OK (S4a)
                                                    3 P55.IK (RDb)
                                                    3 P40.IK (R7b)
                                                    3 P47.IK (RBb)
---- CLK_IN . . . .    TCLKIN.I . . . .    0 GCLK.I
```

A2 IOB PIN ASSIGNMENTS

Queryblk: RAY05A5.LCA (3020PC68-70), XACT 2.12,
Sat Nov 24 15:05:10 1990

I/O Block Assignments for the Dual Correlator

E0a	P12	(.o)	(.t)	(.ok)	(.i)	_aE0	CLK
E0b	P33	(.o)	(.t)	(.ok)	(.i)	_bE0	CLK
E1a	P8	(.o)	(.t)	(.ok)	(.i)	_aE1	CLK
E1b	P41	(.o)	(.t)	(.ok)	(.i)	_bE1	CLK
E2a	P2	(.o)	(.t)	(.ok)	(.i)	_aE2	CLK
E2b	P53	(.o)	(.t)	(.ok)	(.i)	_bE2	CLK
E3a	P65	(.o)	(.t)	(.ok)	(.i)	_aE3	CLK
E3	P49	(.o)	(.t)	(.ok)	(.i)	_bE3	CLK
RAa	P9	(.o)	(.t)	(.ok)	(.i)	_aRA	CLK
RAb	P38	(.o)	(.t)	(.ok)	(.i)	_bRA	CLK
RBa	P7	(.o)	(.t)	(.ok)	(.i)	_aRB	CLK
RBb	P47	(.o)	(.t)	(.ok)	(.i)	_bRB	CLK
RCa	P66	(.o)	(.t)	(.ok)	(.i)	_aRC	CLK
RCb	P51	(.o)	(.t)	(.ok)	(.i)	_bRC	CLK
RDa	P68	(.o)	(.t)	(.ok)	(.i)	_aRD	CLK
RDb	P55	(.o)	(.t)	(.ok)	(.i)	_bRD	CLK
REa	P64	(.o)	(.t)	(.ok)	(.i)	_aRE	CLK
REb	P46	(.o)	(.t)	(.ok)	(.i)	_bRE	CLK
R0a	P20	(.o)	(.t)	(.ok)	(.i)	_aR0	CLK
R0b	P32	(.o)	(.t)	(.ok)	(.i)	_bR0	CLK
R1a	P19	(.o)	(.t)	(.ok)	(.i)	_aR1	CLK
R1b	P31	(.o)	(.t)	(.ok)	(.i)	_bR1	CLK
R2a	P5	(.o)	(.t)	(.ok)	(.i)	_aR2	CLK
R2b	P23	(.o)	(.t)	(.ok)	(.i)	_bR2	CLK
R3a	P17	(.o)	(.t)	(.ok)	(.i)	_aR3	CLK
R3b	P21	(.o)	(.t)	(.ok)	(.i)	_bR3	CLK
R4a	P67	(.o)	(.t)	(.ok)	(.i)	_aR4	CLK
R4b	P22	(.o)	(.t)	(.ok)	(.i)	_bR4	CLK
R5a	P4	(.o)	(.t)	(.ok)	(.i)	_aR5	CLK
R5b	P29	(.o)	(.t)	(.ok)	(.i)	_bR5	CLK
R6a	P16	(.o)	(.t)	(.ok)	(.i)	_aR6	CLK
R6b	P36	(.o)	(.t)	(.ok)	(.i)	_bR6	CLK
R7a	P14	(.o)	(.t)	(.ok)	(.i)	_aR7	CLK
R7b	P40	(.o)	(.t)	(.ok)	(.i)	_bR7	CLK
R8a	P13	(.o)	(.t)	(.ok)	(.i)	_aR8	CLK
R8b	P39	(.o)	(.t)	(.ok)	(.i)	_bR8	CLK
R9a	P6	(.o)	(.t)	(.ok)	(.i)	_aR9	CLK
R9b	P37	(.o)	(.t)	(.ok)	(.i)	_bR9	CLK
S1a	P62	_aSFE1	(.t)	CLK	(.i)	(.q)	(.ik)

```
S1b    P43 _bSFE1 (.t)    CLK       (.i)      (.q)      (.ik)
S2a   P56 _aSFD2  (.t)    CLK       (.i)      (.q)      (.ik)
S2b    P42 _bSFD2 (.t)    CLK       (.i)      (.q)      (.ik)
S4a    P63 _aSFE4 (.t)    CLK       (.i)      (.q)      (.ik)
S4b   P48 _bSFE4  (.t)    CLK       (.i)      (.q)      (.ik)
S8a    P61 _aSFF8 (.t)    CLK       (.i)      (.q)      (.ik)
S8b   P50 _bSFF8  (.t)    CLK       (.i)      (.q)      (.ik)
XINa   P15 (.o)   (.t)    (.ok)     (.i)      _aXIN     CLK
XINb   P24 (.o)   (.t)    (.ok)     (.i)      _bXIN     CLK
XOUTa P3   _aXC   (.t)    CLK       (.i)      (.q)      (.ik)
XOUTb P54 _bXC    (.t)    CLK       (.i)      (.q)      (.ik)
```

A3 CLB Configuration

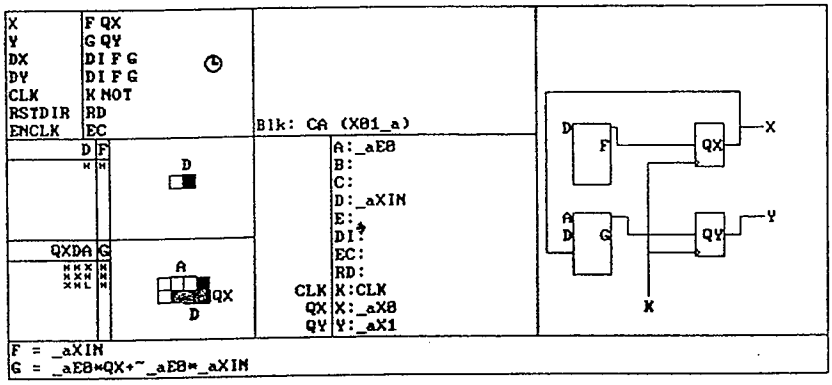


Figure A.1. CLB configured as in figure 3.4 (Shift-Register).

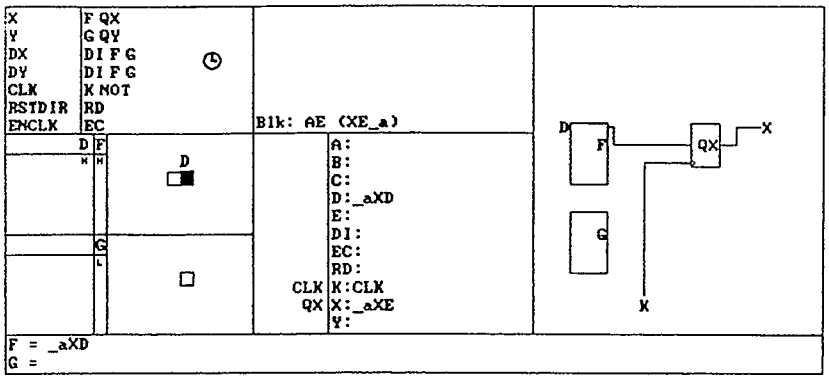


Figure A.2. CLB configured as in figure 3.5 (Shift-Register, last stage).

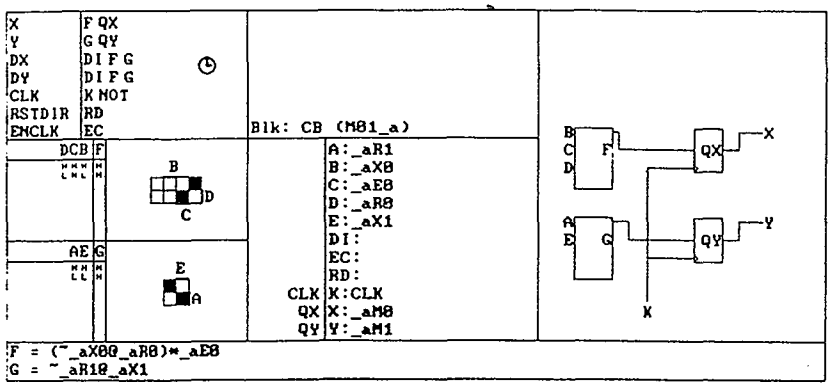


Figure A.3. CLB configured as in figure 3.6 (Comparators).

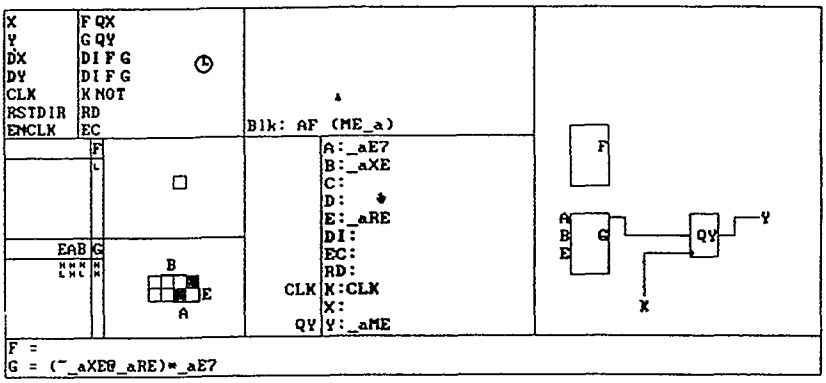


Figure A.4. CLB configured as in figure 3.7 (Comparator, last stage).

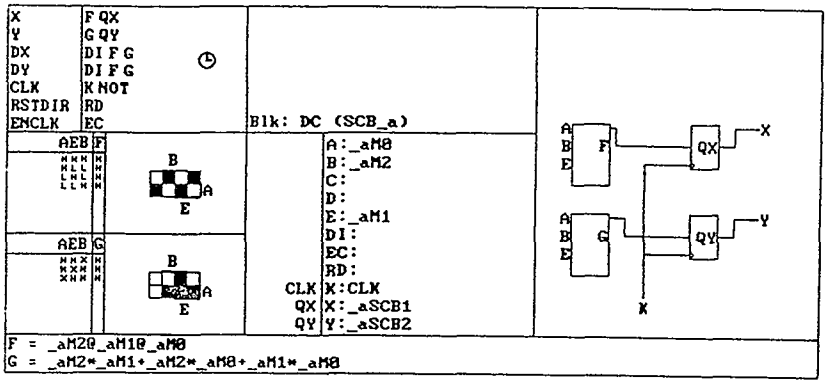


Figure A.5. CLB configured as in figure 3.8. (3-Bit Adder).

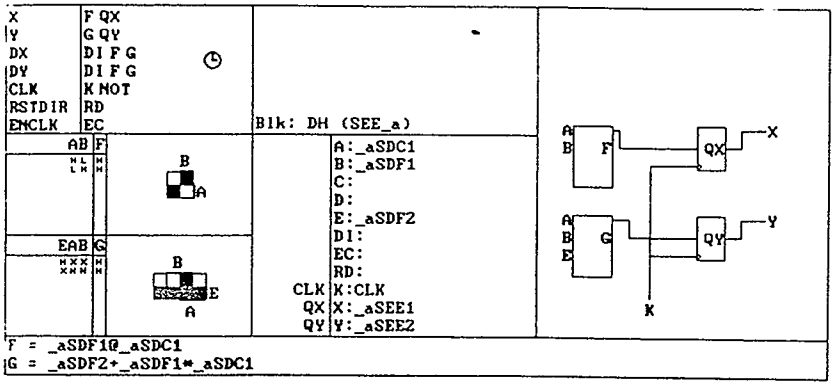


Figure A.6. CLB configured as in figure 3.9. (3-Bit Adder with Carry Look-Ahead).

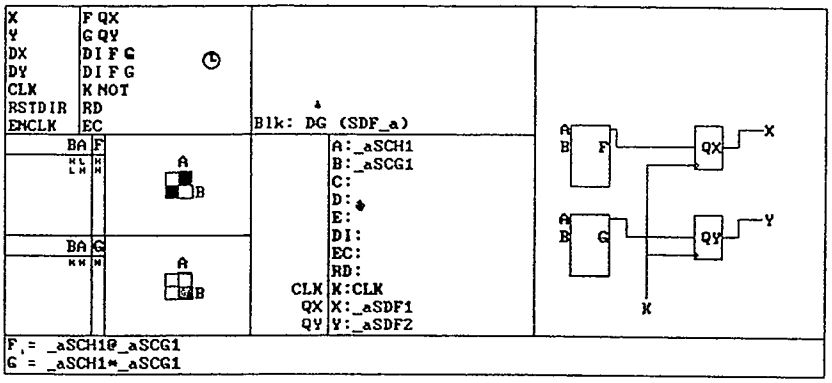


Figure A.7. CLB configured as in figure 3.10 (2-Bit Adder).

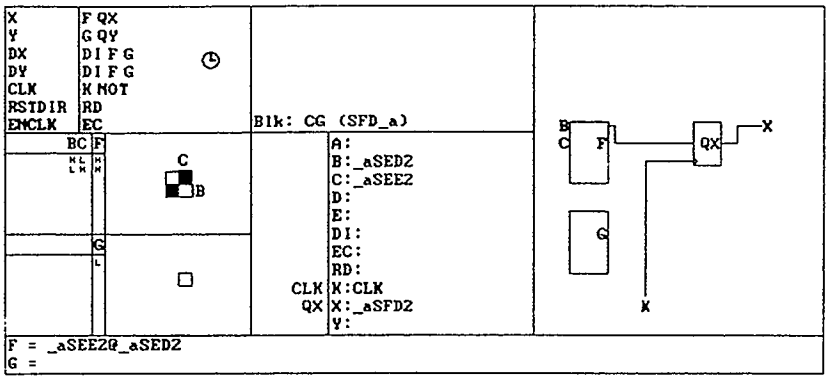


Figure A.8. CLB configured as in figure 3.11 (Half-Adder).

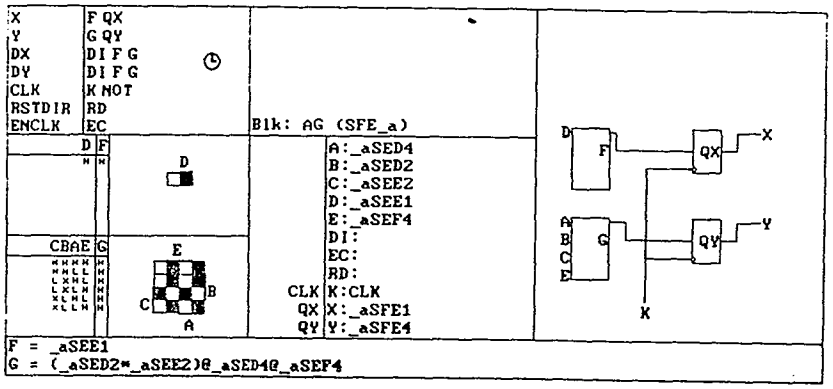


Figure A.9. CLB configured as in figure 3.12 (Adder with Carry Look-Ahead).

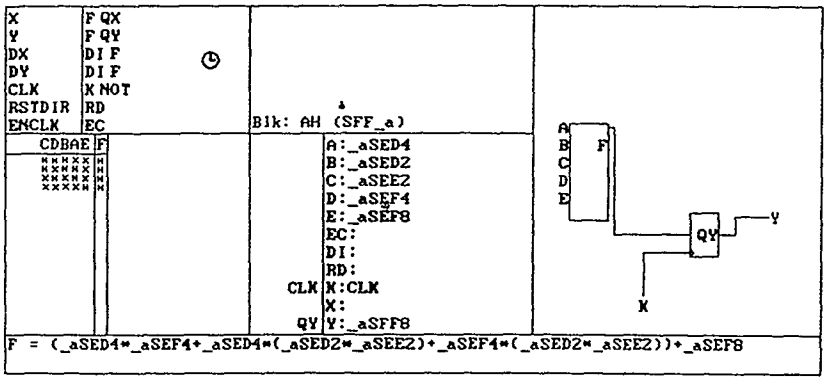


Figure A.10. CLB configured as in figure 3.13 (Adder with Carry Look_Ahead).

WORKS CITED

1. Ziemer, R.E., and R.L. Peterson. **Digital Communications and Spread Spectrum Systems.** New York: Macmillam Publishing Company, 1985.

2. Dixon, R.C. **Spread Spectrum Systems.** New York: John Wiley and Sons, 1984.

3. **The Programmable Gate Array Data Book.** San Jose: Xilinx, Inc., 1888.

BIBLIOGRAPHY

Dixon, R.C. **Spread Spectrum Systems.** New York: John Wiley and Sons, 1984.

Grieco, D.M. "The Application of Charge-Coupled Devices to Spread-Spectrum Systems." **IEEE Transactions on Communications**, VOL. COM-28, NO. 9, (September 1980) :1693-1705.

High-Performance Logic Device Data Sheet: GA22V10 Santa Clara: Gazelle Microcircuits, INC., 1989.

The Programmable Gate Array Data Book. San Jose: Xilinx, Inc., 1888..

TRW LSI Products INC., Data BOOK. La Jolla, Ca: TRW LSI Products INC., 1988.

Turin, G.L. "An Introduction to Digital Matched Filters." **Proceedings of the IEEE**, VOL. 64, NO.7, (JULY 1976): 1092-1112.

Ziemer, R.E., and R.L. Peterson. **Digital Communications and Spread Spectrum Systems.** New York: Macmillam Publishing Company, 1985.