This is to certify that the

dissertation entitled

VLSI Smart Logic Modeling and Design For
Optimum Chip Feature Characterization

presented by

Hsien-Hui Tseng

has been accepted towards fulfillment
of the requirements for

Ph.D. degree in Electrical Engineering

R. Zapp
Major professor

Date 6/30/93

VLSI SMART LOGIC MODELING AND DESIGN FOR OPTIMUM CHIP FEATURE CHARACTERIZATION

By

Hsien-Hui Tseng

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Department of Electrical Engineering

1993

ABSTRACT

VLSI SMART LOGIC MODELING AND DESIGN FOR OPTIMUM CHIP FEATURE CHARACTERIZATION

By

Hsien-Hui Tseng

Because the transportation and handling of agricultural products can result in various degrees of damage due to impacts, a 38 mm (1.5 in.) diameter self-contained Instrumented Sphere (IS) was designed to record the impacts it experiences while being handled with like-sized commodities (plums, tomatoes, peaches, strawberries). Compared with the 89 mm (3.5 in.) diameter sphere previously reported by Zapp et al., 1990, this miniaturized unit contains a composite smart logic chip with the same 32K of memory, approximately 1/12 the volume, smaller batteries, and a smaller triaxial accelerometer.

Spatial optimization was performed by computer to investigate the practical minimum volume configuration that can be realized using commercially available components. The saving in power consumption for the smart logic configuration was also investigated. The combination of reduced power, small circuit elements and optimum packaging produce a feasible sphere of 38 mm (1.5 in.) diameter.

The overall objectives accomplished in this research report are: 1) development of a general design rule for a n-Dimensional (n-D) data acquisition system with m restrictions (m or more of the input signals from each of the n independent sources are above some threshold for signal acceptance); 2) design of a 3-D smart logic data acquisition system with one restriction (This latter system replaces a μ-P circuit and other supporting logic of an operational data acquisition system. Essentially a single chip data acquisition system replaces a circuit board system of 5 chips); 3) demonstration of a VTI smart logic simulation to verify the single chip performance, and; 4) assessment of the size, power consumption, and cost compared to the operational μ-P based system.

To my family

ACKNOWLEDGMENTS

I would like to express my greatful appreciation and gratitude to the members of my Dissertation Committee: Dr. H. Roland Zapp, Dr. Mohammad Aslam, Dr. Galen K. Brown, Dr. Bong Ho, and Dr. David Yen. Their constructive comments and guidance were invaluable to the completion of this thesis.

A special appreciation goes to Dr. H. Roland Zapp and Dr. Mohammad Aslam, my advisors, for serving as chairs of my dissertation committee. Their encouragement, advice, support and friendship helped each step along the way and their excellent dedication provided continuous inspiration.

I wish to acknowledge Dr. Galen K. Brown for providing an assistantship during the course of this project.

I also wish to express my gratitude to Dr. Paul R. Armstrong for his assistance.

At last I wish to express my love and gratitude to my family for their support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Agricultural products, sensitive electronic equipment, glass and bottled goods are among the many items which can be damaged by impact or shock. To identify the sources of damage, instrumentation mounted to the item or substitution of the item by an electronic clone have been proposed. For example, an Instrumented Sphere (IS) accompanying apples during mechanical handling operations has provided the monitoring required to identify locations and magnitudes of significant impacts which cause bruise damage to apples [1],[2]. This IS [3] had a diameter of 89 mm (3.5 in.), which was adequate for measurements with apples, potatoes, cucumbers and other medium sized produce.

The 89 mm (3.5 in.) IS has been very useful for identifying damage causes; impact locations and impact magnitudes. A smaller IS is essential to accomplish similar results in smaller size produce such as strawberries, plums, mushrooms, peaches, eggs, etc.. One method to achieve a size reduction was to re-package the electronics in a 63.5 mm (2.5 in.) diameter IS [4], probably the smallest size possible without the use of very large scale integration (VLSI). A second approach is to develop a multichip housing [5] which replaces four standard integrated circuit (IC) chip with four dice mounted on a single substrate with pinouts compatible to those of the microprocessor (μ-P). A piggyback sandwich type configuration results which has dimensions of just the μ-P. Operational power reduction is necessary to accommodate smaller batteries for the much smaller size IS. Heat sinking must be considered because of the high circuit density in the reduced size. These conflicting requirements suggest a CMOS smart logic VLSI design [6],[7] to achieve the simultaneous objectives of small size and low power consumption.

A CMOS smart logic chip approach should produce the smallest size IS. The approach to chip miniaturization is to eliminate the μ-P and replace it with a "smart logic" low power consuming chip. The motivation here is to simplify the electronics by using a smart logic integrated circuit (SLIC) for performing the system operation instead of the μ-P since the μ-P is highly underutilized in the presently available IS. Thus, by developing dedicated logic, which should be compatible with VLSI technology, the size [8],[9], power consumption [9]~[12] and cost [13] of the miniaturized IS is substantially reduced. In the previous IS designs, the μ-P was the major power consumer. A factor of two in power reduction is realized using smart logic without a μ-P.

A smaller IS which consumes less power than the previous 89 mm (3.5 in.) diameter IS and present 63.5 mm (2.5 in.) diameter is a design objective. A final size of less than 38 mm (1.5 in.) diameter, or about 1/12 the volume of the earlier 89 mm (3.5 in.) model, results. The miniature IS uses a smaller triaxial accelerometer, a SLIC, operational amplifier (Op Amp), analog-to-digital conversion (A/D) and the same 32K of external random access memory (RAM). The final design eliminates the circuit board by incorporating the circuit in a single IC package which contains four IC dice. These dice are mounted on a patterned silicon substrate that fits in the package. After bonding the dice [14]~[17], the package is sealed using standard IC package processing. The completed electronic package is encased in rigid foam and cast in beeswax to provide structural hardness for the IS. After adequate testing, an epoxy casing will be utilized. An optimization routine shows that a 38 mm (1.5 in.) diameter is feasible using commercially available elements. A number of element packaging arrangements were configured with the optimization routine. The results of these optimization configurations are presented. The power and heat dissipation analysis for the smart logic configuration is included.

Chapter 2 gives a review of some researchers' efforts in detecting and explaining bruising in fruit during harvesting and handling. Criteria for 3-Dimension (3-D) data acquisition, VTI and CMOS VLSI design considerations and previous μ-P based data acquisition units are also addressed in that Chapter. Chapter 3 presents system design strategies which include system operation, general design rules and algorithms for n-D data acquisition with m restrictions, special cases and applications and the software listing. Chapter 4 provides the analysis of size, power and heat dissipation for the μ-P based IS, the composite smart logic chip based IS and the absolute size limit for an IS. Power and size are substantially reduced from the commercially available unit and heat dissipation represents no problem for size reduction related to multichip housing. The variation of power and size vs specification changes is also provided. Chapter 5 presents simulation results and discussion. Chapter 6 discusses the general results and conclusions and provides some ideas for future studies.

CHAPTER 2

BACKGROUND

2.1 Historical Perspective

During the past two decades many researchers tried to detect and explain bruising in fruit during harvesting and handling. Rider [18], developed a pseudo-fruit to measure the forces involved in the collisions and to relate these forces to the generation of bruises. In this pseudo-fruit, three separate accelerometers were mounted in a 57-mm diameter steel shell covered with 9.5 mm of Ensolite material. This pseudo-fruit was capable of recording the maximum acceleration and the time duration of an impact by using cables to transfer acceleration signals to a chart recorder. Rider showed that acceleration can be used to predict internal shear stress, if the modulus of elasticity and the radius of the surface impacted by the pseudo-fruit were known. Although physical properties of the Rider pseudo-fruit were similar to those of a peach, the duration of impact did not agree with the values obtained from the Hertz theory. He attributed this to the "bottoming out" of the covering material at relatively high impact values. Rider et al. [19] also studied the pseudo-fruit's calibration and how it would correlate to bruise damage in fruit. The previous other researchers had showed that the probability that a peach will be bruised by impact can be related to the internal shear stress developed in the fruit, based upon the assumption that the fruit behaves elastically. So Rider calibrated the accelerometer outputs to the shear stress experienced by the pseudo-fruit. Thus the shear stresses experienced by the pseudo-fruit were supposed to correlate directly to bruise damage. All of Rider's formulas assumed perfectly elastic impacts. In order to test the mathematical relationships, a 76.2 mm (3.0 in.) diameter, 204 gm (0.45 lb) pseudo-fruit which contained a piezoelectric triaxial accelerometer was constructed. The small flexible coaxial cables were connected between this encased sensor and a tape recorder. The shell of the pseudo-fruit was a 1.52 mm (0.06 in.) thick 57.15 mm (2.25 in.) diameter steel sphere covered with a layer of 9.53 mm (3/8 in.) thick type AH Ensolite^{R} over which three layers of 3M* Fastbond-10^{R} contact cement were applied. The accelerometer was rigidly mounted inside the shell. The unit had a coefficient of restitution of 0.42 when dropped 152.4 mm (6 in.) onto concrete. The modulus of elasticity of the pseudo-fruit, 489 kPa (71 psi), was determined both by measuring the area of contact during impact and by using a quasi-static compression test. In Rider's calibration procedure the only externally supplied variable, was the modulus of elasticity of the impacted surface. From the data, he only made use of the peak acceleration and impact duration. In order to make the theoretical calibration procedure correspond to experimental data however, the impact duration time had to be multiplied by an unexplained factor of 2 before being used in the calibration procedure.

Pullen and Diener [20] developed a low cost FET triaxial accelerometer. The advantages of this system were the vector summing capabilities, using long cables without cable noise or cross talk, high power gain and low impedance output. Three cables were required to transmit the impact data to a recorder. This accelerometer approach was subsequently duplicated by others in recent telemetry systems. Another approach was to use telemetry in a pseudo-fruit to transmit the variation in accelerations via a FM transmitter. The objective was to remove any sensing cables from the pseudo-fruit in order to enable the pseudo-fruit to simulate the free movement of a fruit.

O'Brien et al. [21] developed a telemetry system which contained three miniature and separate FM transmitters with antennas, one for each axis and a triaxial accelerometer. The accelerometer and electronics were enclosed in a 50.8 mm (2 in.) hollow fiber glass sphere covered by 10.2 mm (0.4 in.) thick layer of resilient material. A standard FM receiver was used to pick up the pseudo-fruit signal and store this signal on a multi-channel analog tape recorder. Because of low transmitter power, the receiver's antenna had to be placed very close to the pseudo-fruit. The calibration accuracy was estimated to be ±5%. The pseudo-fruit was not sensitive to accelerations below 0.5 g (gravitational force), or to frequency response down to 2 Hz, because these were believed unimportant for bruise formation.

Aldred and Burch [22] developed an electronic shock-sensing device for detecting amplitudes and frequencies of acceleration of peaches during harvesting and handling. A microcomputer and an analog vector summing circuit were added to this impact detection system. This impact detection system still consisted of a sensing unit and receiving station. Instead of transmitting 3 channels of data, only one transmitter (for the resultant acceleration) was placed inside a 67 mm (2.64 in.) sphere for transmitting one channel of data. Data from the sensor were telemetered to a microcomputer, then processed, stored and retrieved later as a direct readout of maximum impact or as a curve on a strip chart recorder. Approximately 2 sec of data at a 1 kHz sampling rate were stored and processed for each impact. It was stated that additional experimentation was necessary to insure that the pseuso-fruit resembles a real fruit and to correlate output of the pseudo-fruit with the actual bruise. No attempts were made to calibrate the output of this system.

Jenkins and Humphries [23] developed a new technique to assess impact damage. They used a fluid filled bladder with slit valves to meter fluid flow during impact. This bladder was a hollow vinyl sphere (toy ball) approximately 76 mm (3 in.) in diameter with a 3 mm (0.12 in.) wall fitted with six equally spaced one-way slit valves. According to the relationship between water loss and impact velocity, this system resulted in good correlation between drop height on a hard surface and water loss. The only drawback was the lack of an automated recording system.

Halderson et al. [24] developed their first generation impact detection telemetry system to predict impact bruises to potatoes. They used a triaxial accelerometer configuration with three miniature transmitters which operated in the commercial FM band and a three channel FM receiver. The unit operated up 30 m (100 ft.)from the receiver but was too directional. This research, however, lacked both a true calibration procedure and a field test performance analysis. Their second generation impact detection device had a single transmitter system with three subcarriers. The most efficient and non-directional antenna for the unit were two wire loops of approximately 6 cm (2.4 in.) diameter which oriented 90 degrees to each other. A special three channel FM telemetry receiver was used to receive the transmitted signal. During impact tests, the correlation of the enclosed accelerometer voltage with decorder voltage was slightly better than 80%. By 1986 Halderson et al. [25] had built and tested a third generation impact detection device. The main changes from his previous device were in the packaging. The new unit enclosed all of the electronics in a 40 mm x 40 mm x 57mm aluminum box. Three small LC antenna were mounted on the outside of the three perpendicular planes of the box. The aluminum box was molded in silicone (Dow Corning RTV-3110) to form a cylindrical package that was 100 mm (4 in.) in length and 84 mm (3.3 in.) in diameter, weighed 654 gm (1.44 lb), and had an overall specific gravity of 1.18. The device was tested under impact conditions by dropping a 1054 mm (41.5 in.) in length, 286.7 gm (0.63 lb) in weight metal rod, with a spherical hard rubber tip (36.5 mm (1.44 in.) in diameter), onto the device which was supported by a 75 mm (3 in.) thick foam pad with a force-deflection rate of 275 (gm/cm^2)/cm (9.93 (lb/in.^{2})/in.). Ten replications of 0.2 J (1.9 x 10^{-4} BTU) of energy were used for each axial direction. Coefficients of variation were 8.3%, 8.4% and 5.2% respectively for the X, Y and Z axes. The transmission distance was evaluated around a potato harvester, but no range distances were reported.

Anderson and Parks [26] developed two impact detection devices (two physical units) using a pressure sensor transducer in one and a single axis accelerometer in the other. The pressure version was packaged by mounting the electronic assembly inside a hollow rubber ball which was then sealed, thus containing air at atmosphere pressure. The accelerometer version is built up by surrounding the electronic assembly with cushioning foam and an outer shell. Both devices used telemetry to transmit data to a receiver with an attached tape recorder. A two channel tape recorder was used so that data could be recorded on one channel and voice commentary on the other. Both devices used the same FM transmitter design and battery configuration which could operate for eight hours. The pressure version was calibrated by compressing it between two plates with a known force, while the accelerometer version was calibrated by dropping it from known heights. The acceleration data was used to generate an "equivalent drop height number". These devices were used to test potato handling equipment.

Kerr and Wilkie [27] described a triaxial accelerometer and telemetry system, as an automated data collection system, for use on a potato harvester to provide an immediate indication of damage. This unit included a computer on-board the harvester which stored the acceleration, temperature and other critical data for further analysis. No calibration, accuracy or performance results were discussed, however.

More recent work was performed by Siyami [28], Zapp and Armstrong [3], Sober [2] and Brown [1]. The strong continuing interest is due to the recent developments in VLSI design and implementation techniques and the continuing development of miniaturized accelerometers and batteries. The recent technical advances have also reduced the power consumption of most components.

2.2 Previous μ-P Based Data Acquisition Units

The application of the μ-P based system to data acquisition systems has been common.

Lowther et al. [29] designed a general purpose dual-microprocessor data acquisition system. The hardware allowed data capture at speeds of up to 600 kHz and the software provided the user with a simple but powerful interface for setting up test requirements. These requirements included a certain amount of real time control for retrieving and examining the data. This system has been used to investigate the penetration of magnetic fields into nonlinear laminated media under transient and steady state conditions and also to study an electromagnetic suspension system and linear motor drive for advanced ground transportation research.

Hill et al. [30] developed a μ-P based digital wattmeter which can measure power in the frequency range dc to 1 kHz with a full scale accuracy of better than 0.5%. First, the μ-P measured the voltage waveform period and then computed the sampling interval and number of samples. The measurement of average power must be taken over an integer number of cycles. Second, it is necessary to multiply together the samples of the voltage and current waveforms, accumulate the products over the measurement period, and then divided by the number of samples. Finally, we can read out as a 4-digit display of the scaled average power.

Wallingford [31] designed a simple data acquisition scheme which was implemented on a 16-bit microcomputer. This system can simultaneously acquire and store the output of two independent 8-bit A/D at a 115 kHz rate with a 3 MHz clock. This high speed data acquisition is made possible by configuring the interface to respond to two separate nonflicting parallel processes. The simultaneous conversion of two A/D's made this data acquisition scheme well suited for FFT signal processing systems.

Sridharan [32] developed a synchronous multichannel (8-channel) data acquisition system by using a separate A/D converter for each channel. The synchronous sampling became very desirable when the data was required for system identification studies, or when fast data conversion of a large number of analog channels was involved. For a microcomputer clock period of 330 nsec, the program took only 93 μsec of CPU time to convert and store the results of 8 channels in RAM.

Adam [33] made a telemetric seismic data acquisition system which sampled transducers continuously but stored data only when a threshold was exceeded. This system consisted of remote encoding stations and data acquisition stations. Each remote station consisted of up to three seismic sensors operating at sampling rates of 60 samples/sec. The highest single sensor sampling rate was 240 samples/sec. If any of 24 possible encoder stations registered above the threshold for a predetermined time, all of the encoder stations were recorded. If all of the signals were below the threshold for a short period of time, recording was terminated. In order to make the threshold process immune to nonseismic disturbance such as rain and eliminate the need for threshold adjustments, it is necessary to divide the short term average by the long term average.

Ahrens et al. [34] developed a multi-channel microcontroller-based data acquisition unit for logging the activity of cattle on the range which was small enough to be carried by cattle without bothering them. The chewing and walking habits of the cattle were of primary interest to the researchers. Significant motion from these habits produced 5 volt pulses out of the sensor and conditioning circuits which could be sent to the microcontroller.

Negro [35] designed a low power μ-P controlled data acquisition system which used nonvolatile bubble memory cartridges for mass data storage. The system was battery powered. By chosing magnetic bubble memory, the period of unattented operation in the field can increase from 2 weeks to about 2 months and improve reliability. The input voltage to the data acquisition system is derived from the output of a gamma-ray ionization chamber that uses a temperature-compensated electrometer, or other moderately high output transducer. CMOS technology and power switching techniques were used in this system achieving very low power consumption (less than 10 mW).

2.3 Criteria for 3-D Data Acquisition

The need for data acquisition is very common in industrial applications. Some examples of user needs have already been introduced in Section 2.2. For a n-D data acquisition system, in order to save memory by recording only useful data, some restrictions or thresholds for data acceptance are predetermined. For example, in a 3-D data acquisition system for measuring and storing impact accelerations along 3 orthogonal axis, a vector threshold or a single axis threshold can be set in order to efficiently utilize the available memory. The single axis restriction is more easily implemented than a vector restriction. The n-D and m restrictions on a data acquisition system means discarding n data components if less than m of these components are above some threshold where $n \geq m$. For example, if $n=4$, and $m=2$, with each threshold=5, any 2 or more of the 4 dimensions which have absolute values above 5 results in storing all four components. If less than 2 components are above 5, all four components are discarded. The criteria for the 3-D and 1 restriction data acquisition system include the capability of recording and saving 3 data components if any 1 or more of these components are greater than some threshold, for instance ±10 g acceleration. This algorithm can be realized as follows: if $|x(n)| \geq 10$ or $|y(n)| \geq 10$ or $|z(n)| \geq 10$, then output=x, y, z, otherwise, output=0, 0, 0. This provides a transfer function given by $H(z)=1$ when $|x(n)| \geq 10$ or $|y(n)| \geq 10$ or $|z(n)| \geq 10$, otherwise $H(z)=0$.

2.4 VTI & CMOS VLSI Design Considerations

VTI is a comprehensive VLSI design technique [36]~[48] from VLSI Technology, Inc. It covers a broad range of chip design tasks, including behavioral modeling, schematic entry, simulation, symbolic layout, hand-crafted layout, analysis, and test description.

Complementary Metal Oxide Silicon (CMOS) technology has played an increasingly important role recently because of the very low power consumption by this design configuration. Two types of MOS field effect transistors (FET) are produced by this technology, an n-channel MOSFET (NMOS) and a p-channel MOSFET (PMOS). The NMOS transistor consists of n diffusion, polysilicon, metal and insulating layers, while the PMOS transistor consists of p diffusion and the remaining similar materials.

The CMOS inverter, which uses an NMOS transistor as the driver and a PMOS transistor as the load, is characterized by low power consumption in the quiescent state, since one of the two series transistors is always off except during a switching transition from one logic level to the other. Since power dissipation is a concern for VLSI NMOS chips, CMOS is an attractive alternative for VLSI application. For this reason we use VTI tools, which include CMOS standard cells, to design the smart logic chip.

CHAPTER 3

SYSTEM DESIGN STRATEGIES

3.1 A Review of the μ–P Based IS

The existing μ–P based IS consists of one circuit board. The main electronic components consist of an 8-bit CMOS Motorola μ–P (XC68HC11) which has an integral 8-channel A/D converter, 8K byte ROM (which stores the monitor program), and a RS232 serial communication port. The other electronic components include 32 K RAM, latch, multiplexer and Op Amp.

The analog signal flow originates from a triaxial piezoelectric accelerometer (Columbia model 512TX) and adjacent conditioning circuits. The conditioning circuits provide impedance matching, voltage range scaling and noise filtering before input to the 0-5 Vdc A/D on-board the μ–P. The μ–P provides a multiplexed 8-channel 8-bit A/D converter with sample and hold.

The IS power supply consists of a rechargable, 7.2 V NiCad battery (Eveready CH22) chosen because of its availability, reliability, cost and size. The charge capacity is about 80 mA-h. Voltage regulation to 5 Vdc is accomplished with a high efficiency low differential regulator from National Semiconductor (LM2931AZ5). When the IS is in operation the current drain is less than 14 mA, which corresponds to approximately 6 h of battery power. At lower sampling rates or in a "sleep" mode the current can be reduced substantially (down to 2 mA in the sleep mode).

The entire design is enclosed in a 89 mm (3.5 in.) diameter sphere cast in wax to reduce the construction interface problem.

3.2 Smaller Size and Reduced Power Consumption by Smart Logic

The application of digital electronics to data acquisition systems is common. A sophisticated µ-P based data acquisition system for agricultural damage monitoring has been proposed and demonstrated. But the µ-P occupies a relatively large volume and demands the most system power. Thus, a new design idea using smart logic without a µ-P and other auxiliary components such as a latch and a multiplexer would be desirable. The new design concept should accomodate a number of desired features, such as small size and long operation time. The design incorporates a suitable clock, a selection algorithm to choose the correct analog signals, and a threshold filter. A memory address counting scheme to store desired digital data is also required. The absence of the µ-P assures a reduction in power consumption by a factor of 2. However, without the µ-P, the system will suffer some flexibility; for example, the sampling rates will not be changable, the data dimensions and thresholds can not be altered and the programability is lost. These features are not critical for a dedicated data acquisition device. Each new application will require custom designed acquisition features.

3.2.1 General Design Rules for n-D Data Acquisition with m Restrictions

A general data acquisition unit allowing n data lines and m restrictions is developed. This general design can be reduced to the specific case to be realized in hardware. The n-D and m restrictions means saving n data components if any m or more of these components are greater than some threshold where $n \geq m$. For example, n=4, m=2, threshold=5, if any 2 or more of the 4 components have absolute values greater than 5 all 4 data components are recorded otherwise these components are ignored. The design concept requires developing a suitable clock, latch and other auxiliary circuits to realize a low cost, low power consumption single chip data acquisition system. In order to get a suitable clock, as the specification requires, it is necessary to develop a clock design. After obtaining the desired clock, it is still necessary to choose the correct analog input signals for the A/D, so that the appropriate binary code development is necessary to multiplex the correct input signal. It is also necessary to threshold the signals from the A/D, to provide signal latching and to generate a comparison circuit with a threshold, to identify the desirable signals. Finally, we need a 16-bit counter for address counting of the memory. The design specification is divided into the following five main parts:

3.2.1.1 Clock Design

A CSC (Counter & Switch Clock generator) circuit can generate a t μsec clock burst (t represents the μsec duration of the clock pulses in the burst) with f MHz frequency (f represents the frequency of the burst), see Figure 1, for input to the CTB (clock-to-binary generator), as shown in Figure 2. In addition, the CSC generates a similar signal, when latch enable (LE_) is low, for input to a 16-bit counter, identified as counter16.

Assuming that the crystal provides a clock period of t′, and considering the A/D specifications, we can choose t such that

$t / t' = p = 2^q$

where q is an integer. Thus we need a q bit counter, identified as counter1, which converts an input clock period t′ from the crystal into an output period t, see Figure 3. Every f MHz cycle, we will keep n (data dimension) clock pulses of t μsec duration each and ignore the (1/tf) - n remaining clock pulses. We will call this waveform SWC (switch clock), as shown in Figures 1 and 2. The waveform SWC is generated as follows:

Express 1/tf (clock pulses count per sampling frequency) as a binary code $d_{15}d_{14}$

$\ldots d_2 d_1 d_0$. If 1/tf is not an integer, it should be rounded off to form an integer. Define $f'=(1/t)\cdot(1/(d_{15}d_{14}\ldots d_l\, d_{l-1}d_{l-2}\ldots d_1d_0))$, so that if 1/tf is an integer then $f'=f$, otherwise $f' \approx f$.

In our circuit design, we need the lower l bits of the sequence $d_{15}d_{14}\ldots d_l d_{l-1}d_{l-2}\ldots d_1d_0$ equal to n in order to get n clock pulses of t μsec duration per f MHz cycle as shown in Figure 1, where n is the data dimension and l=round-off of $\log_2 n+1$.

Define $s_{l-1}\ldots s_0=n$ where n, the data dimension, equals the decimal equivalent to the binary sequence. If $d_{l-1}d_{l-2}\ldots d_1d_0$ equals $s_{l-1}\ldots s_0$, let f'', the real frequency in the SWC waveform, be equal to $(1/t)\cdot(1/(d_{15}d_{14}\ldots d_2d_1d_0))$ or $f''=f'$. In the general case where $d_{l-1}d_{l-2}\ldots d_1d_0$ does not equal $s_{l-1}\ldots s_0$, we can replace $d_{l-1}d_{l-2}\ldots d_1d_0$ by $s_{l-1}\ldots s_0$ according to the following round off rules:

If $d_{l-1}d_{l-2}\ldots d_1d_0 > s_{l-1}\ldots s_0$,

or $s_{l-1}\ldots s_0 - d_{l-1}d_{l-2}\ldots d_1d_0 \leq 2^{l-1}$, then $f''=(1/t)\cdot(1/(d_{15}d_{14}\ldots d_l\, s_{l-1}\ldots s_0))$;

whereas, if $s_{l-1}\ldots s_0 - d_{l-1}d_{l-2}\ldots d_1d_0 > 2^{l-1}$, then $f''=(1/t)\cdot(1/(d_{15}d_{14}\ldots d_l\, s_{l-1}\ldots s_0-2^l))$.

The above rules will result in a minimun error of $f'-f''$ where f'' is the real frequency in the SWC waveform and f' is approximately equal to f. If $f'=f$ and $f'-f''$ gives zero, the actual frequency will be the ideal specified frequency.

In order to generate the SWC signal a second counter, identified as counter2, is necessary to accept clock t and output $Q_{15}Q_{14}\ldots Q_1Q_0$, see Figure 3. From the $Q_{15}Q_{14}\ldots Q_l$ sequence choose those which have a value of 1 when the corresponding modified values (consistent with previously discussed requirements) of $d_{15}d_{14}\ldots d_l$ has a 1 and connect these Q values to an AND gate. After $(1/f'')$ - tn μsec the output of the AND gate, T1, will go high to enable the SWC. Similarly, from the $Q_{15}Q_{14}\ldots Q_1Q_0$ sequence choose those which have a value of 1 when the corresponding modified values of $d_{15}d_{14}\ldots d_2d_1d_0$ has a 1 and connect these lines to a NAND gate. For this latter case, we need to add an extra t' to the NAND gate input to avoid a transient spike. After $1/f''$ μsec the output of the NAND gate, T2, goes low, which sends a low reset, RS1, to counter2 to reset $Q_{15}Q_{14}\ldots Q_1Q_0$.

This allows T1 to go low, so that the total high time is tn. The resulting T1 and t generate the SWC used in the CTB. From SWC and LE_, we can get CNTC (counter clock) for counter addressing. See Figure 3 for the sequence described above.

Example:

Assume n=3, m=1, t=2 μsec and t′=125 nsec. Then $t/t'=16=2^4$. As seen in Figure 3, counter1 receives input from the clock with t′=125 nsec and sends out a clock with t=2 μsec. In the design considered here, n=3, f=0.003, so every 0.003 MHz we keep 3 clock pulses and ignore the remaining 164 clock pulses in order to obtain 3 clock pulses per 0.003 MHz, each of 2 μsec duration. This waveform is shown in Figure 4, and is called SWC.

The waveform SWC is generated as follows:

The number of pulses per cycle is expressed by its binary representation:

$1/tf=1/(2\ \mu sec \times 0.003\ MHz) \approx 167=10100111=d_7d_6\ldots d_2d_1d_0$

Since n=3, or $s_1s_0=11$, and d_1d_0 is the same as s_1s_0, the sequence $d_7d_6\ldots d_2d_1d_0$ =10100111 remains unchanged.

The counter2, shown in Figure 3 receives 2 μsec clock pulses and outputs $Q_7Q_6\ldots Q_1Q_0$, from which $Q_7Q_5Q_2$ can be choosen for input to an AND gate. After 328 μsec (2 μsec x $(2^7+2^5+2^2)$), T1 (the output of the AND gate) will go high to enable the SWC. Similarly, from the $Q_7Q_6\ldots Q_1Q_0$ sequence choose $Q_7Q_5Q_2Q_1Q_0$ and connect to a NAND gate. For this latter case, we need to add an extra 125 nsec clock pulse to the NAND gate in order to avoid a transient spike. Without this delay clock pulse, the output from the NAND gate will induce a 4.1 nsec spike due to a timing conflict with counter2. This spike will produce a 3.0 V signal at RS1, which will incorrectly enable counter2 and will generate an ambiguous output $Q_7Q_6\ldots Q_1Q_0$. Assuming correct timing, after 334 μsec (2 μsec x $(2^7+2^5+2^2+2^1+2^0)$), T2, the output from the NAND gate, will go low and send this low, RS1, to reset all output $Q_7Q_6\ldots Q_1Q_0$ of counter2, so that T1 goes low giving a total high time of 6 μsec. Using T1 and a 2 μsec clock, we can generate SWC to be used in the CTB. From SWC and LE_, we can get CNTC for counter addressing.

3.2.1.2 Binary Code Development

The A/D has n channels (for example, the National Semiconductor ADC0808 data sheet shows 8 channels), which are multiplexed by analog switches to choose the appropriate analog input. It is necessary to develop the binary code for correct A/D addressing. This is accomplished by the CTB circuit, as shown in Figure 5, which accepts the pulse sequence SWC from the output of the CSC, and generates the binary code $s_{l-1} \ldots s_0$, where l=round-off of $\log_2 n + 1$and n is the data dimension for A/D addressing.

3.2.1.3 Signal Latching

Because we need to threshold filter the signals from the A/D, a latch is necessary to store these signals in a buffer memory. The latch operates in two stages; First, we need to store n parallel input signals from the A/D into a pre-latch every f MHz cycle in order to investigate m constraints. Second, if m or more constraints are satisfied, a low LE_ is sent to the CSC to control the clock for the counter16 and pass the n signals into the post-latch. The data is stored in memory during the next f MHz cycle. If the constraints are not satisfied, the LE_ will remain high and the post-latch will block the signal.

3.2.1.4 Comparison of Input Signal with a Threshold

The signal from the A/D is threshold filtered in order to save memory. Thus, we require an 8 bit comparator circuit to compare the signals from the A/D with predetermined thresholds (both positive and negative thresholds). If a signal satisfies the threshold, a low LE$_i$_ ($1 \le i \le n$) is sent to the latch. If the thresholds are not satisfied, the LE$_i$_ will remain high. Thus only interesting data will be latched into memory.

3.2.1.5 Memory Address Counting

The memory has a 16-bit address bus, so a 16-bit counter is necessary to receive the clock pulses from the CSC and send the 16 bit count signal to memory for address counting.

The block diagram for this n-D data acquisition with m restrictions circuit is shown in Figure 2.

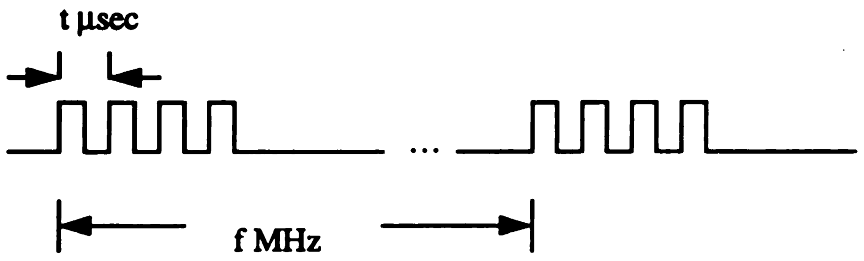


Figure 1 - A t μsec clock with f MHz frequency forms the waveform SWC.

(n=4, l=round-off of $\log_2 n + 1 = 3$, $s_{l-1} \ldots s_0 = s_2 s_1 s_0 = 100 = n$)

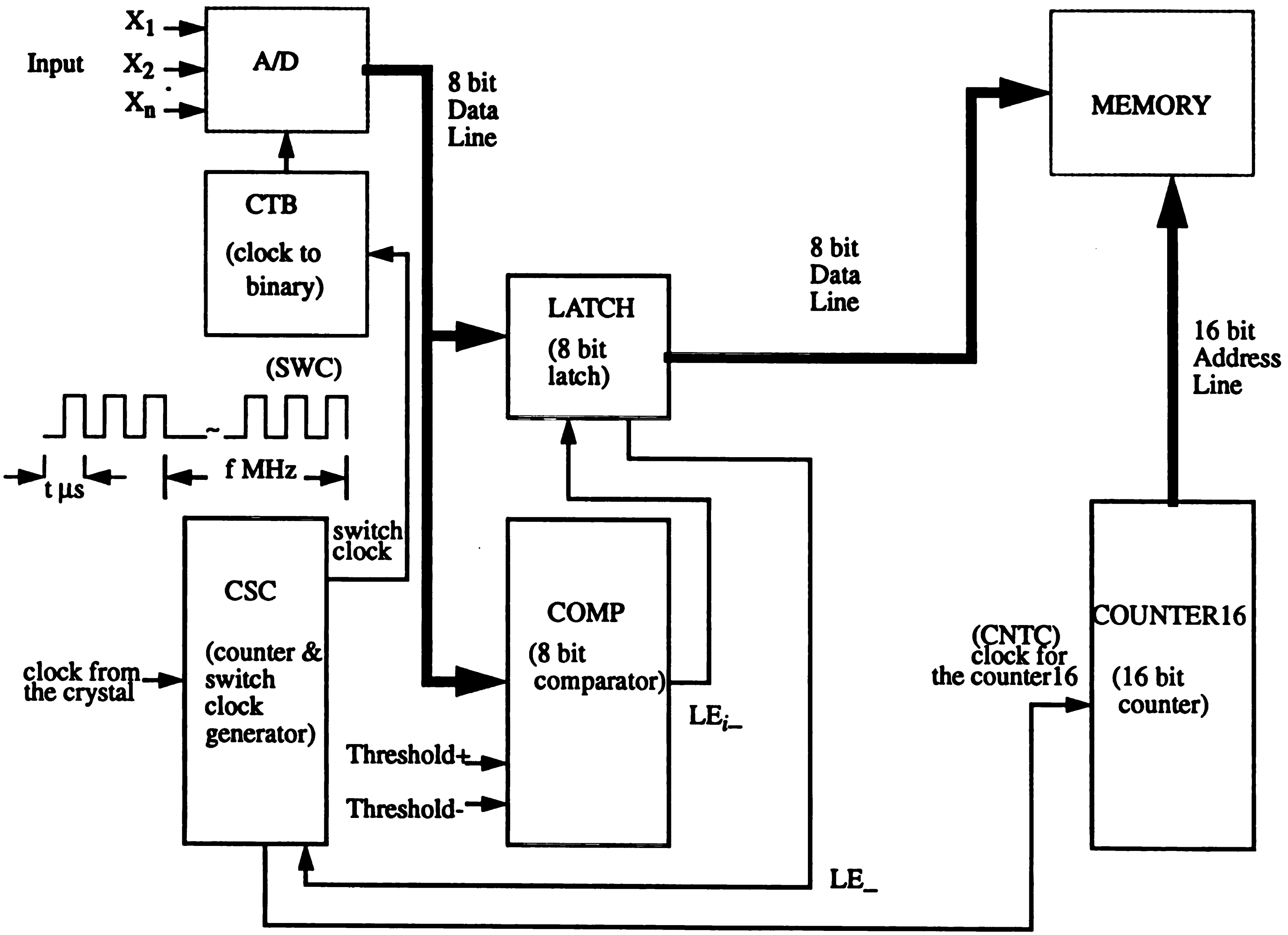


Figure 2 - Block diagram for the n-D data acquisition system with m restrictions.

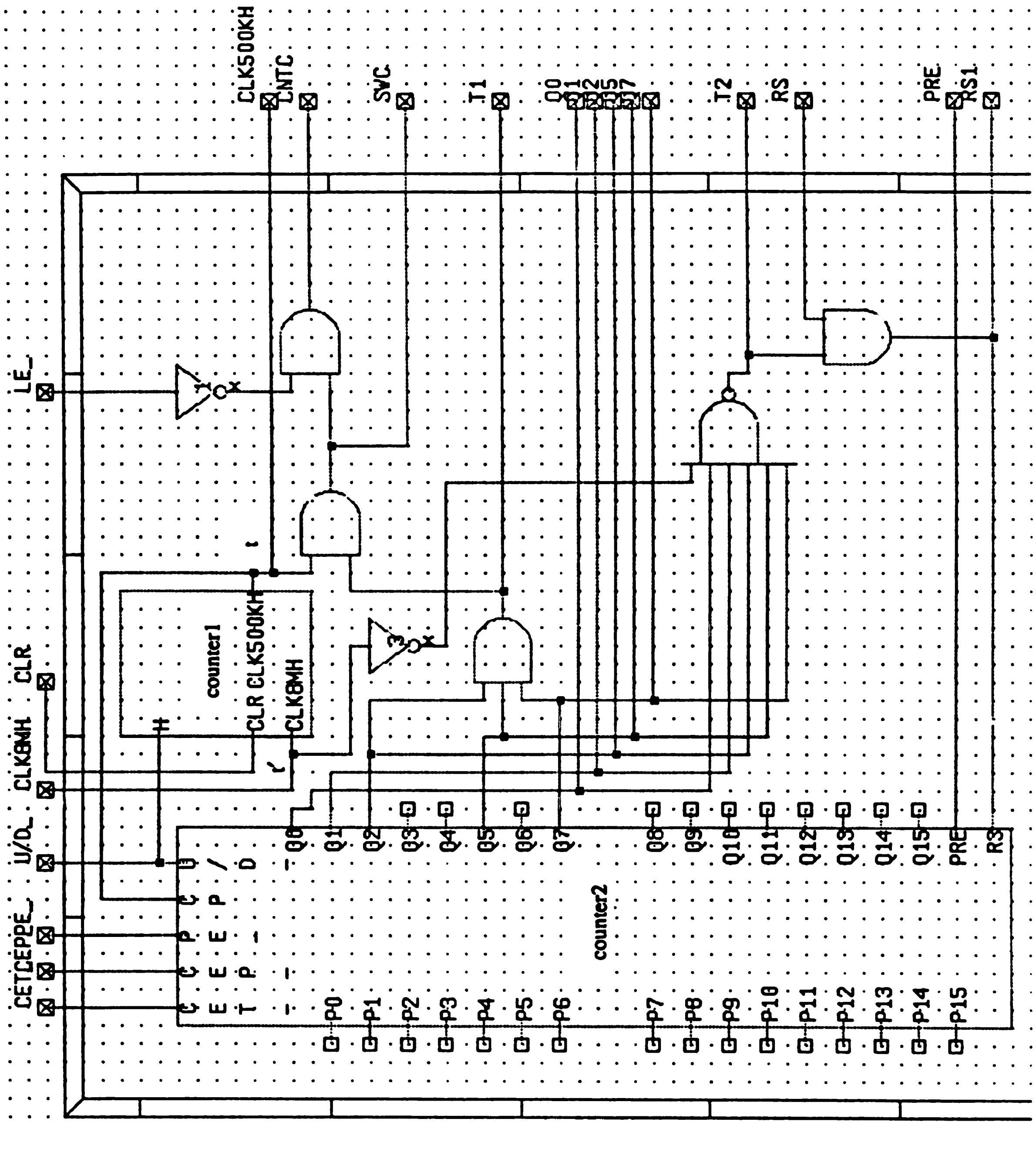


Figure 3 - The detailed circuit diagram for the CSC (counter & switch clock) generator for n=3.

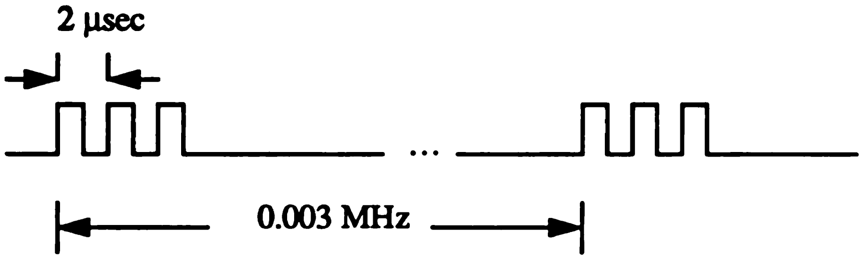


Figure 4 - The waveform SWC, with 3 pulses of 2 μsec at a 0.003 MHz repetition frequency.

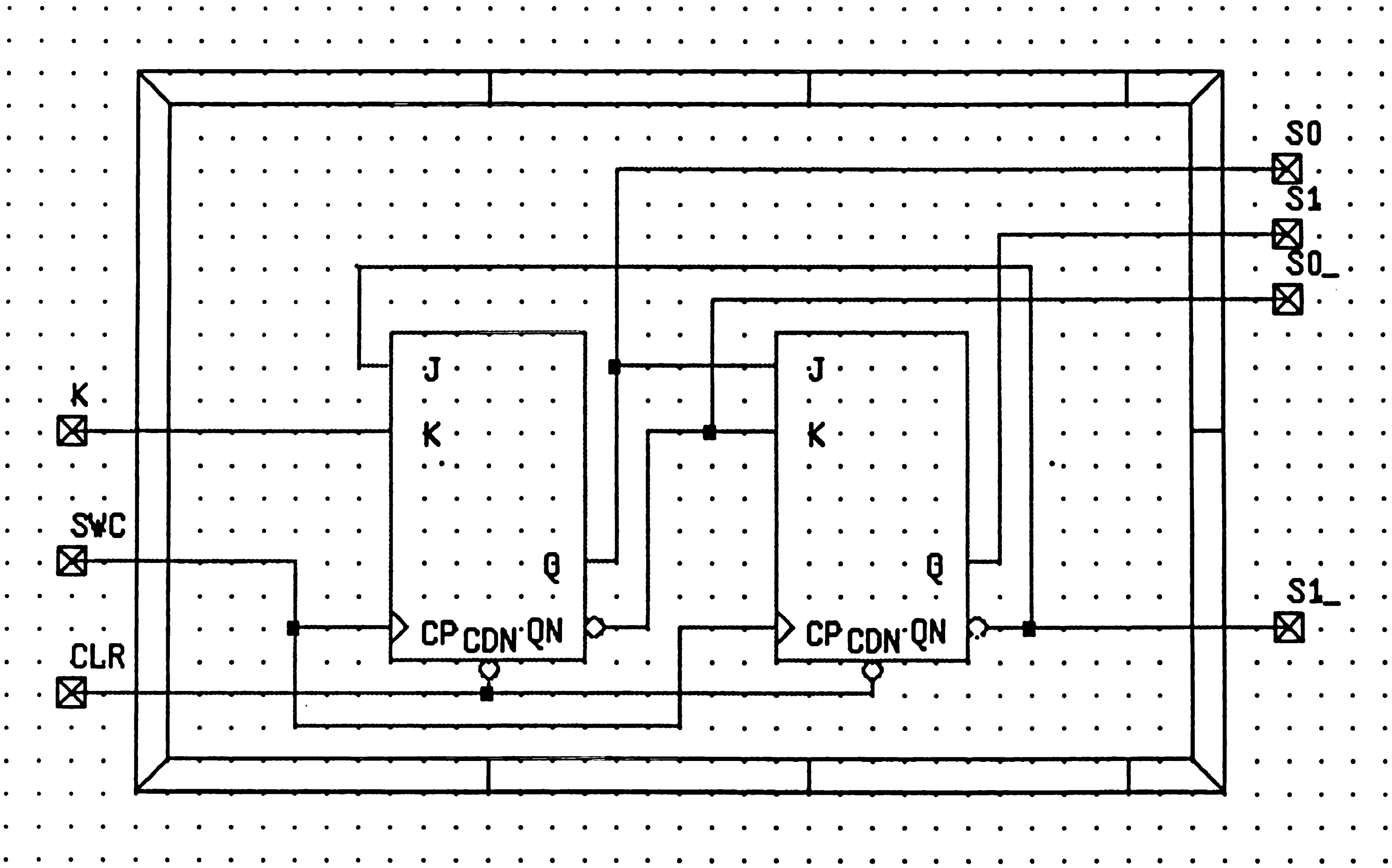
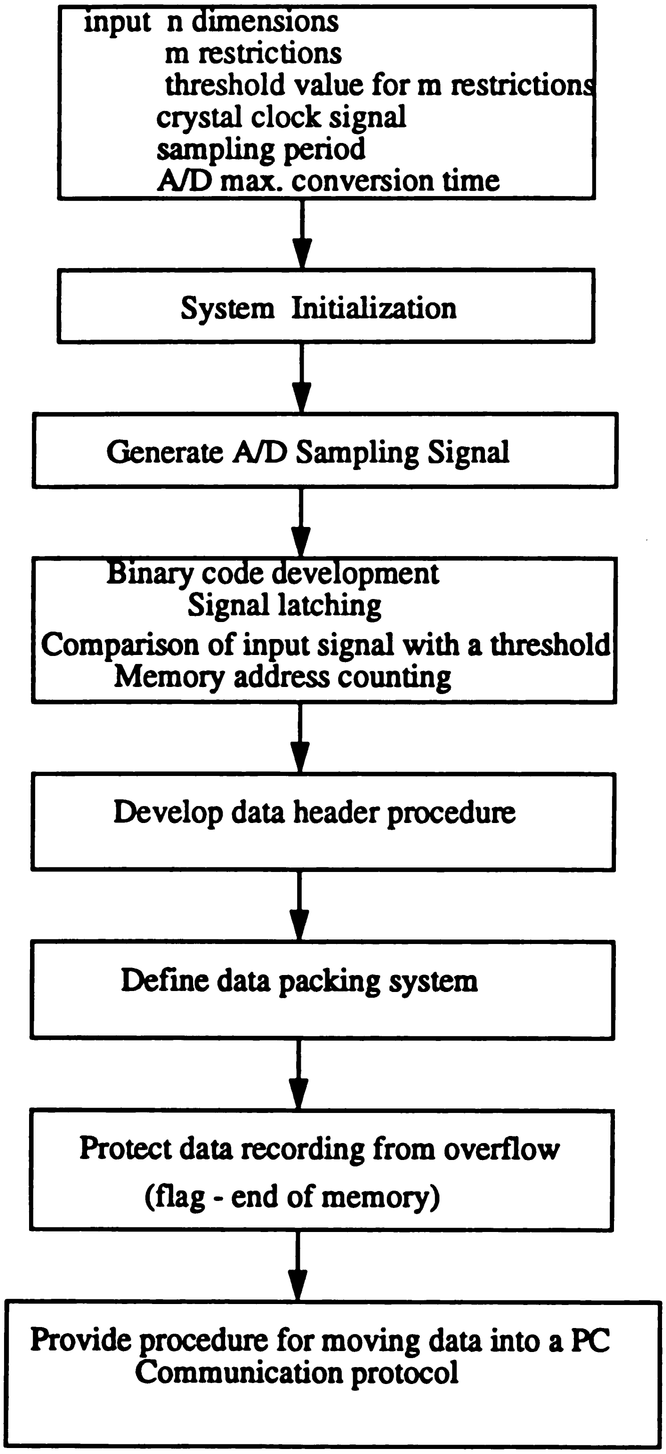


Figure 5 - The detailed circuit diagram for the CTB (clock-to-binary) generator for n=3.

3.2.2 Flow Chart for System Operation

An overall flow chart for system operation is shown as follows:

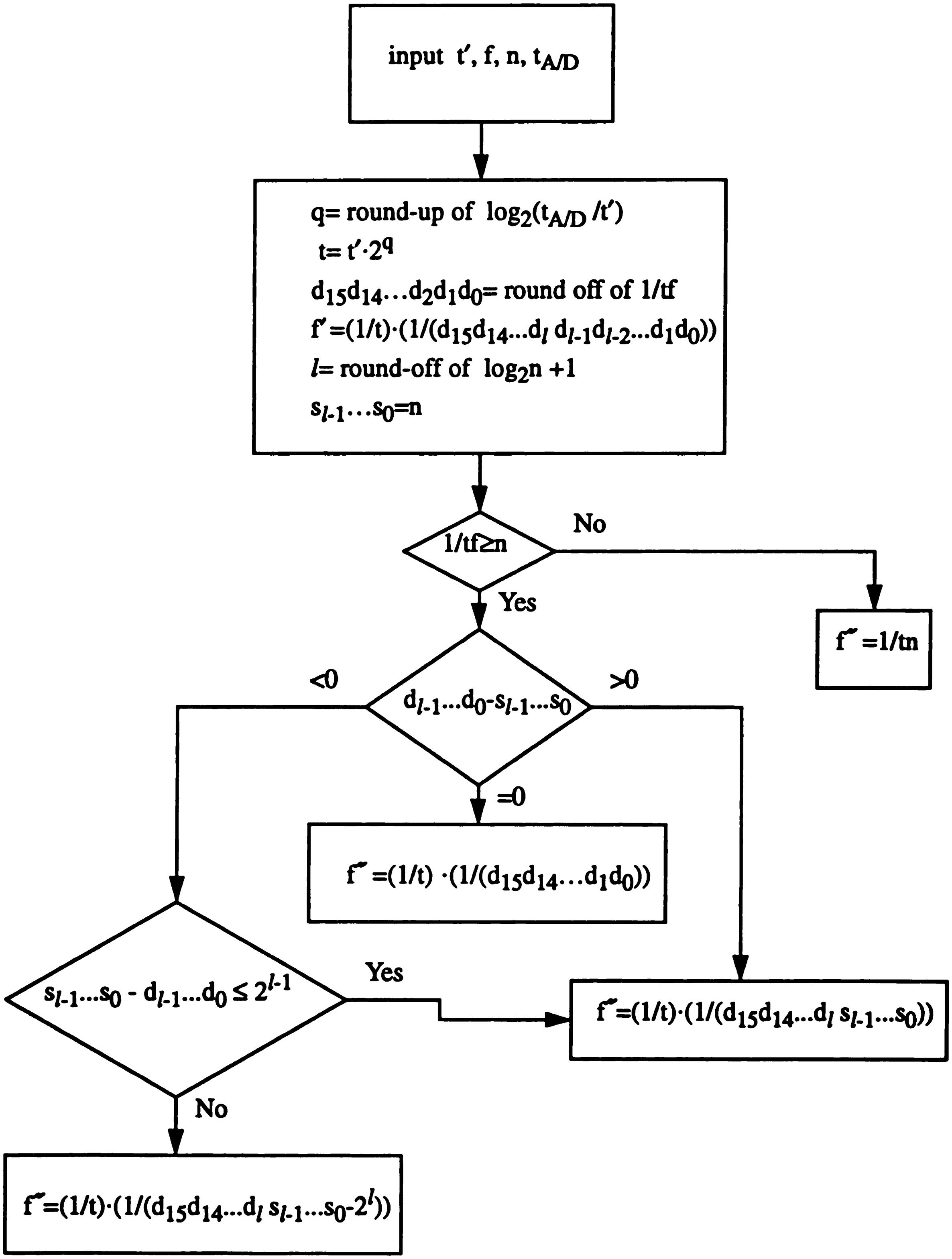

input n dimensions
m restrictions
threshold value for m restrictions
crystal clock signal
sampling period
A/D max. conversion time
System Initialization
Generate A/D Sampling Signal
Binary code development
Signal latching
Comparison of input signal with a threshold
Memory address counting
Develop data header procedure
Define data packing system
Protect data recording from overflow
(flag - end of memory)
Provide procedure for moving data into a PC
Communication protocol

3.3 The Algorithm

Develop a new design methodology [49]~[53] to synthesize a n-dimension with m restrictions data acquisition system which includes the previously defined CSC, CTB, latch, comparator and counter16 for optimum power and size features.

The development of the overall data acquisition signals will procede as outlined below. The design methodology follows the flow graph attached below to develop the actual sampling frequency $\tilde{f}$, with minimun error in the difference f- $\tilde{f}$.

Flow graph for developing the actual sampling frequency f'' with minimun error in the difference f- f''.


input t', f, n, $t_{A/D}$
q= round-up of $\log_2(t_{A/D}/t')$
$t = t' \cdot 2^q$
$d_{15}d_{14}\ldots d_2d_1d_0$= round off of 1/tf
$f' = (1/t)\cdot(1/(d_{15}d_{14}\ldots d_l\, d_{l-1}d_{l-2}\ldots d_1d_0))$
l= round-off of $\log_2 n + 1$
$s_{l-1}\ldots s_0 = n$
$1/tf \geq n$
No
Yes
$f'' = 1/tn$
<0
>0
$d_{l-1}\ldots d_0 - s_{l-1}\ldots s_0$
=0
$f'' = (1/t)\cdot(1/(d_{15}d_{14}\ldots d_1d_0))$
$s_{l-1}\ldots s_0 - d_{l-1}\ldots d_0 \leq 2^{l-1}$
Yes
$f'' = (1/t)\cdot(1/(d_{15}d_{14}\ldots d_l\, s_{l-1}\ldots s_0))$
No
$f'' = (1/t)\cdot(1/(d_{15}d_{14}\ldots d_l\, s_{l-1}\ldots s_0 - 2^l))$

3.4 Special Case (Clock Count Per Burst Less Than Data Dimension)

When $1/tf < n$ where t, f, and n were previously defined, and $t / t' = p = 2^q$ (with q an integer and t' as previously defined), we still need a $q+l$ bit counter, counter1, which receives an input clock period t' from the crystal and generates the waveform SWC, see Figure 6. For $f'=1/tn$, the actual frequency from the SWC is generated as follows:

From the $Q_{q+l-1}Q_{q+l-2}\ldots Q_q$ sequence choose those which have a value of 1 when the corresponding values of $d_{q+l-1}d_{q+l-2}\ldots d_q$ has a 1, where $d_{q+l-1}d_{q+l-2}\ldots d_q$ is the binary code of n, and connect these lines to a NAND gate. Again, we need to add an extra t' to the NAND gate input to avoid a transient spike. After 1/f µsec the output of the NAND gate, T2, goes low, which sends a low reset, RS1, to counter1 to reset $Q_{q+l-1}Q_{q+l-2}\ldots Q_qQ_{q-1}\ldots Q_1Q_0$. The Q_{q-1} ($t'x2^q=t$) is the SWC used in the CTB. From SWC and LE_, we can get CNTC for counter16 addressing, see Figure 7 for this sequence.

3.5 Application (Design of a Smart Logic 3-D Data Acquisition System with One Restriction)

According to the general design rules, with n=3, and m=1, we can achieve the design of a smart logic circuit to replace an operational µ-P based system. The design will contain the following five main blocks: CSC, CTB, LATCH, COMP, COUNTER. The design procedure follows the techniques outlined for the n-D data acquisition with m restrictions. The five required blocks are as follows:

3.5.1 CSC Design

The counter and switch clock generator generates the desired signals for the CTB and for the counter16. In our case, the input clock t' from the crystal is 125 nsec. According to the A/D specification the maximum conversion time is 116 µsec (National Semiconductor

ADC0808). In the design considered here, n=3, f=0.003, so that 1/tf=1/(116x0.003) ≈2.87<3=n, which indicates that these conditions belong to the special case design considered in Section 3.4 above. If we choose t=128 μsec, then t/t'=128/0.125 =1024=2^{10}, or q=10, and with l=round-off of $\log_2 3$ +1=2, we need a 12-bit counter (q+l), counter1, which can be the cascade of a first stage 4-bit counter, counterA, and a second stage 8-bit counter, counterB. CounterA receives its input from the clock with period 125 nsec (from the crystal) and generates an output Q_3 of period 2 μsec. CounterB receives the 2 μsec clock signal from Q_3 of CounterA and generates the waveform SWC, see Figure 8. If we define f''=1/tn=1/(128x3) ≈0.002604 as the real frequency for the SWC, the waveform for SWC is generated as follows:

From the $Q_7Q_6...Q_1Q_0$ sequence of counterB choose Q_7Q_6 (2 μsec x (2^7+2^6)=384 μsec= 128 μsec x3) and connect these lines to a NAND gate along with an extra 125 nsec clock pulse in order to avoid a transient spike. After 384 μsec the output of the NAND gate, T2, goes low, which sends a low reset, RS1, to counterB to reset $Q_7Q_6...Q_1Q_0$. The Q_5 (2 μsec x 2^6=128 μsec) of counterB is the SWC used in the CTB. From SWC and LE_, we can get CNTC for counter16 addressing, see Figure 8 for this sequence.

The simulation results of the CSC generator circuit for the special case n=3, m=1, t'=125 nsec, $t_{A/D}$ =116 μsec, f=0.003 MHz are shown in Figure 9. Each of the monitoring nodes can refer to the circuit diagram shown in Figure 8. The SWC and CNTC are expected signals. CLK8MH (t') and LE_ are input signals. At first LE_=0, RS is set to low to clear counterA and forces RS1 to 0 which clears $Q_7Q_6...Q_1Q_0$ of counterB. Then RS is set to high and forces RS1 to 1, counterA and $Q_7Q_6...Q_1Q_0$ of counterB counts step by step according to the clock sequence of CLK8MH and the output of counterA, respectively. T2 is an important signal, which is initially high and after 384 μsec (2 μsec x (2^7+2^6)=384 μsec= 128 μsec x3) goes low and sets RS1=0 in order to reset $Q_7Q_6 ...Q_1Q_0$ of counterB. The Q_5 (cycle time of one pulse=2 μsec x 2^6=128 μsec) of counterB is the SWC (contains 3 pulses of Q5) used in the CTB. From SWC and LE_, we can get CNTC for counter16 addressing. Then T2 goes high and sets RS1=1 to continue a new cycle of SWC.

3.5.2 CTB Design

The A/D has 8 channels (National Semiconductor ADC0808) for multiplexing different analog inputs. The clock-to-binary circuit accepts 3 pulses of 128 µsec signals at 0.002604 MHz repetition frequency, see Figure 10, converts these to binary codes s_1s_0 and sends these codes to the A/D for choosing x, y, z axis analog inputs.

The simulation results of the CTB generator circuit for n=3 are shown in Figure 11. Each of the monitoring nodes can be referred to the circuit diagram shown in Figure 5. The S1 and S0 are CTB output control signals for choosing respectively the x, y, z axis analog A/D input signals. The SWC is an input signal representing a series of bursts from the CSC. The system initialization is as follows: the K of the JK Flip Flop is always set high, the CLR (clear) is set low in order to set S1 and S0 to 00, after which CLR is set high. The S1 and S0 change from 00, 01, 10 to correctly select the x, y, z axis analog inputs respectively and return to 00 according to the clock count of the SWC. The cycle continues in order to choose the desired signal. Each cycle needs three pulses from the SWC. S1_ and S0_ are the complements of S1 and S0, respectively, and these four signals are used in the latch for selecting digital signals from three different axes.

3.5.3 LATCH Design

The latch circuit accepts three signals from the A/D at a 0.002604 MHz rate, and generates a LE_ (latch enable low true) signal which is sent to the CSC to control the CNTC. For each 0.002604 MHz cycle, if one or more of the received signals are above the threshold, the low enable latch will strobe the signals into memory. For signals below the threshold, the LE_ will remain high and the latch will block these signals from memory.

The simulation results for the latch circuit for the special case n=3, m=1, t'=125 nsec, $t_{A/D}$ =116 µsec, f=0.003 MHz are shown in Figure 12. Each of the monitoring nodes are shown in the circuit diagram of Figure 13. The O7,O6,......,O0 and LE_ are output signals. The former is sent to memory as input data and the latter is send to the CSC to control the CNTC. The LE_{i}_ (*i*=x,y,z, referred to as I8 in the simulation), I7,I6,.....,I0 are input signals representing the latch enable signal of a single axis and digitalized impact force, respectively. The signals shown on Figure 12, namely, SWC, RS1, S1 and S0 are control signals. The input signal used to run a simulation is an impact force similar to that shown in Figure 14. The output signals are delayed one time unit (384 μs) and the first two data values are replaced by a heading and time of occurrence for any significant impact pulse. The output signals are updated only when SWC is high. S1 and S0 are used to choose the input signals from the x, y, z axis, where S1,S0=00 chooses the x axis, S1,S0=01 chooses the y axis, and S1,S0=10 chooses the z axis. The S1,S0 control signals continue to cycle in order to select the desired signal from the data sequence.

In Appendix 1 and Figure 14, the first input data to the latch for the three axes is 84H, 84H, 84H, which is below the threshold. The relative output data which is delayed one time unit is undefined, and in particular resulted in the sequence: 80H, 80H, 84H for our simulation. The second input data is 84H, 84H, C0H, which is above the threshold. The relative output data, which is delayed one time unit, is 80H, 80H, 80H, which represents heading. The third input data is E0H, E0H, E0H, which is also above the threshold. The relative output data, which is delayed one time unit, is 00H, 00H, 0CH, which represents the real time clock count. The fourth input data is E0H, E0H, 84H, which is above the threshold. The relative output data, which is delayed one time unit, is E0H, E0H, 84H, the same as input data. The fifth input data is 84H, 84H, 84H, which is below the threshold. The relative output data, which is delayed one time unit, is arbitrary for example E0H, 84H, 84H in the simulation performed in this thesis.

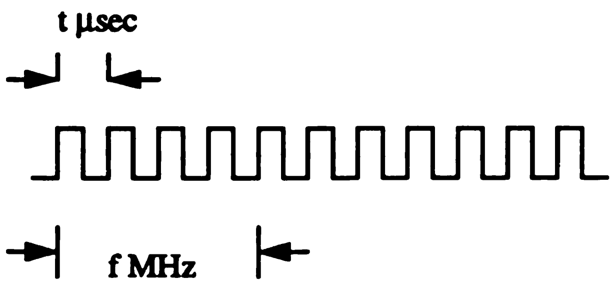


Figure 6 - A t μsec clock with f MHz frequency forms the waveform SWC.

(for the special case and n=4)

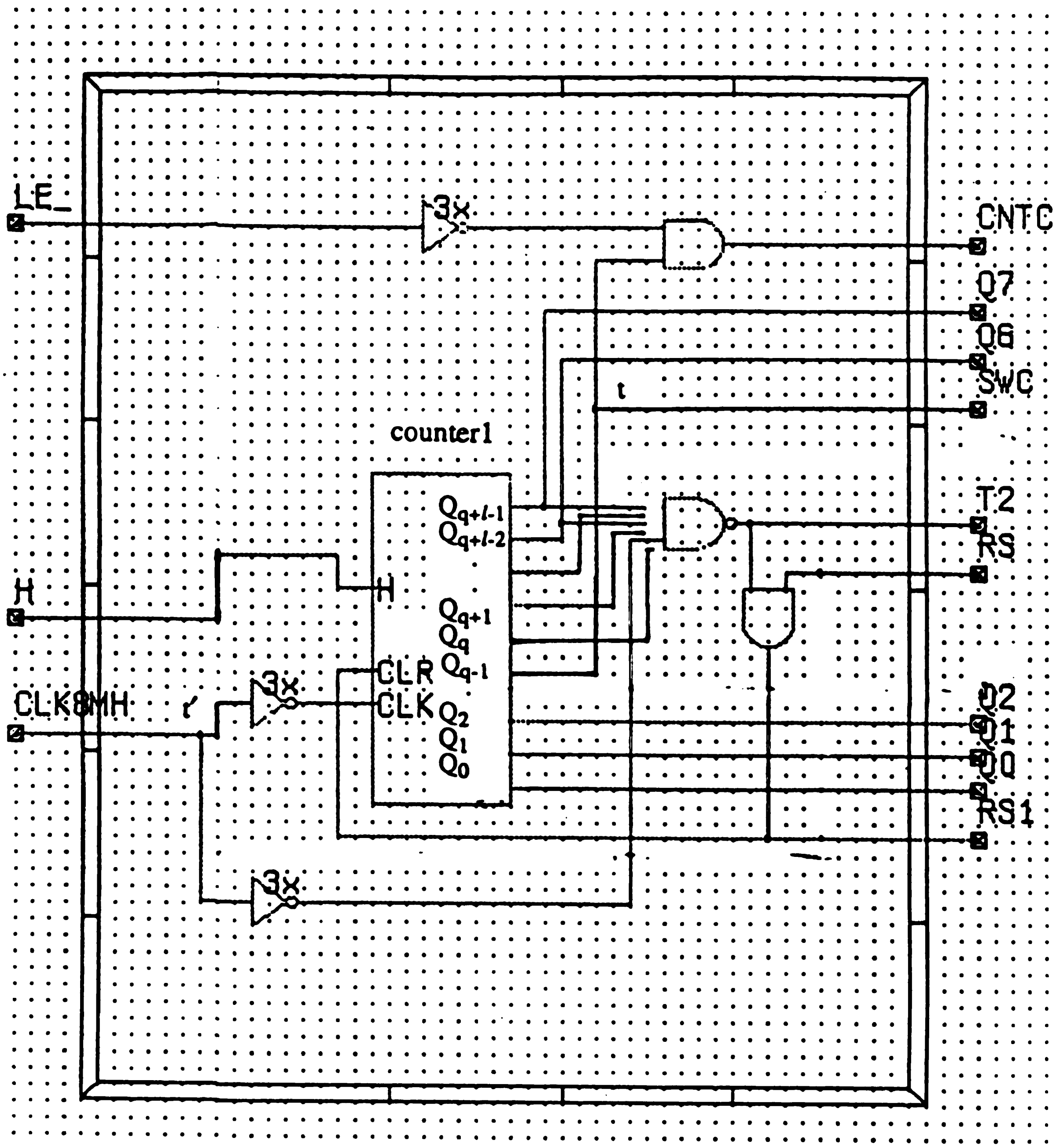


Figure 7 - The schematic circuit diagram for the CSC (counter & switch clock) generator for the special case n=3.

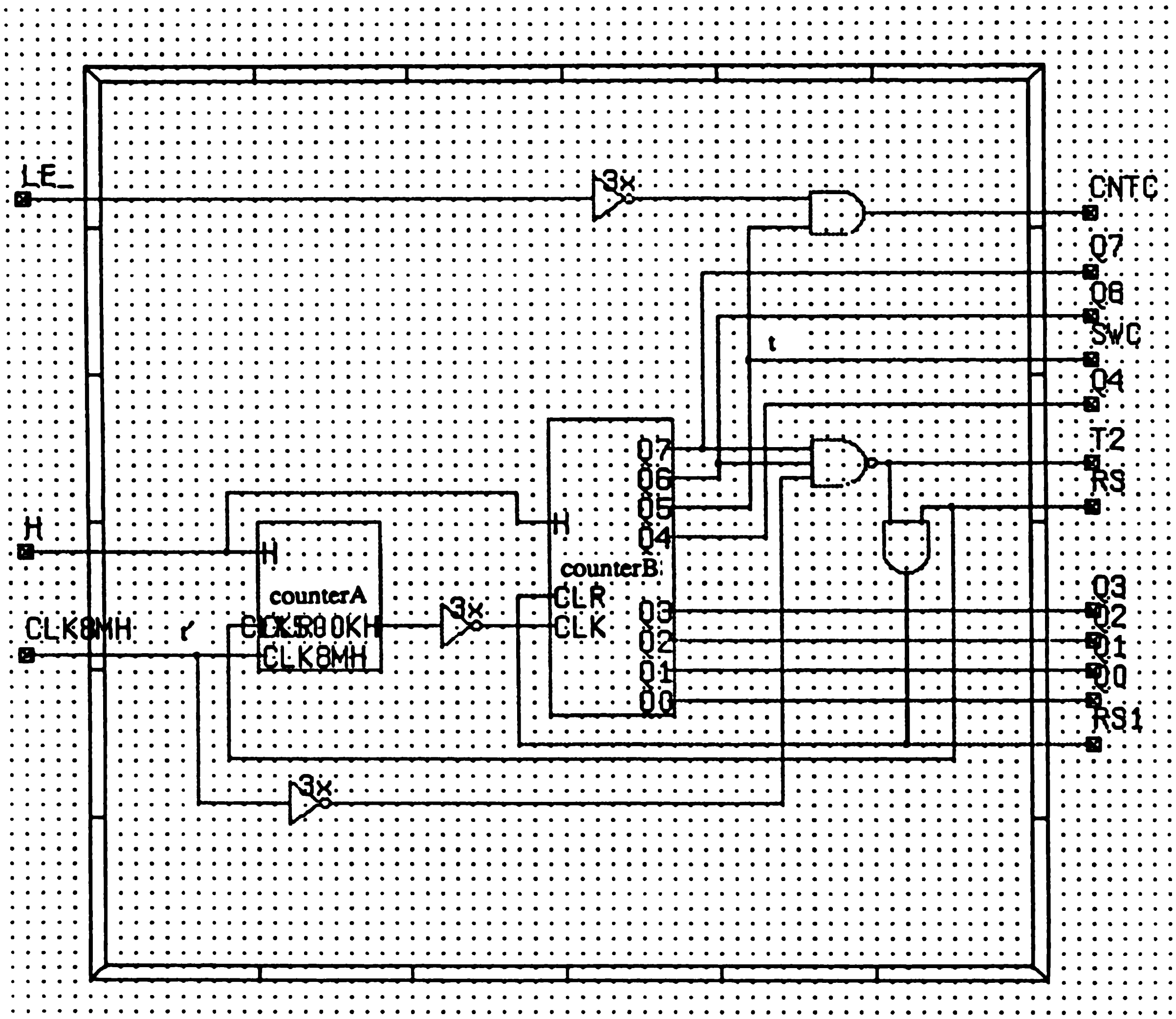


Figure 8 - The detailed circuit diagram for the CSC (counter & switch clock) generator for the special case n=3.

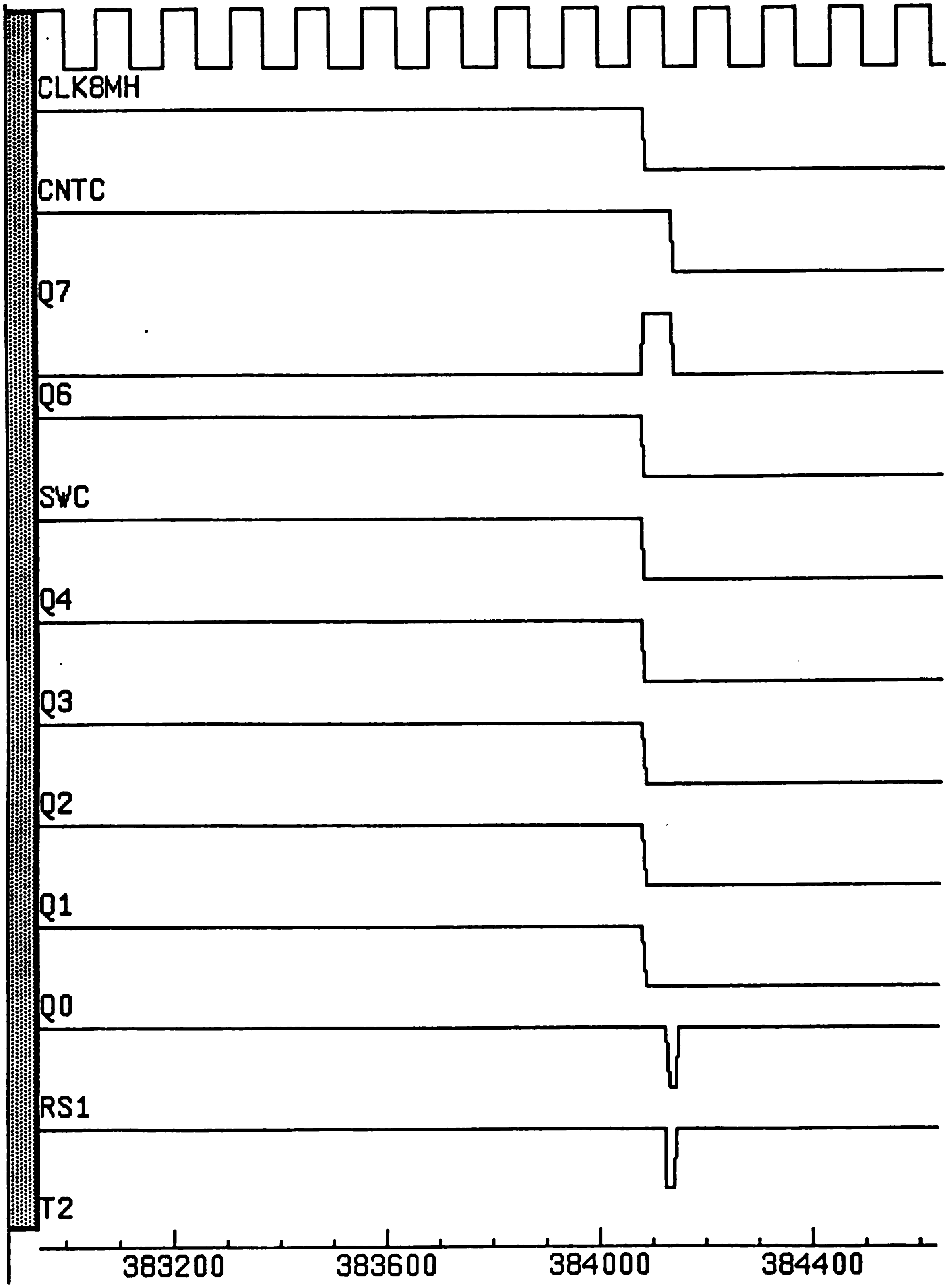


Figure 9.a - The simulation results of the CSC (counter & switch clock) generator circuit for the special case n=3.

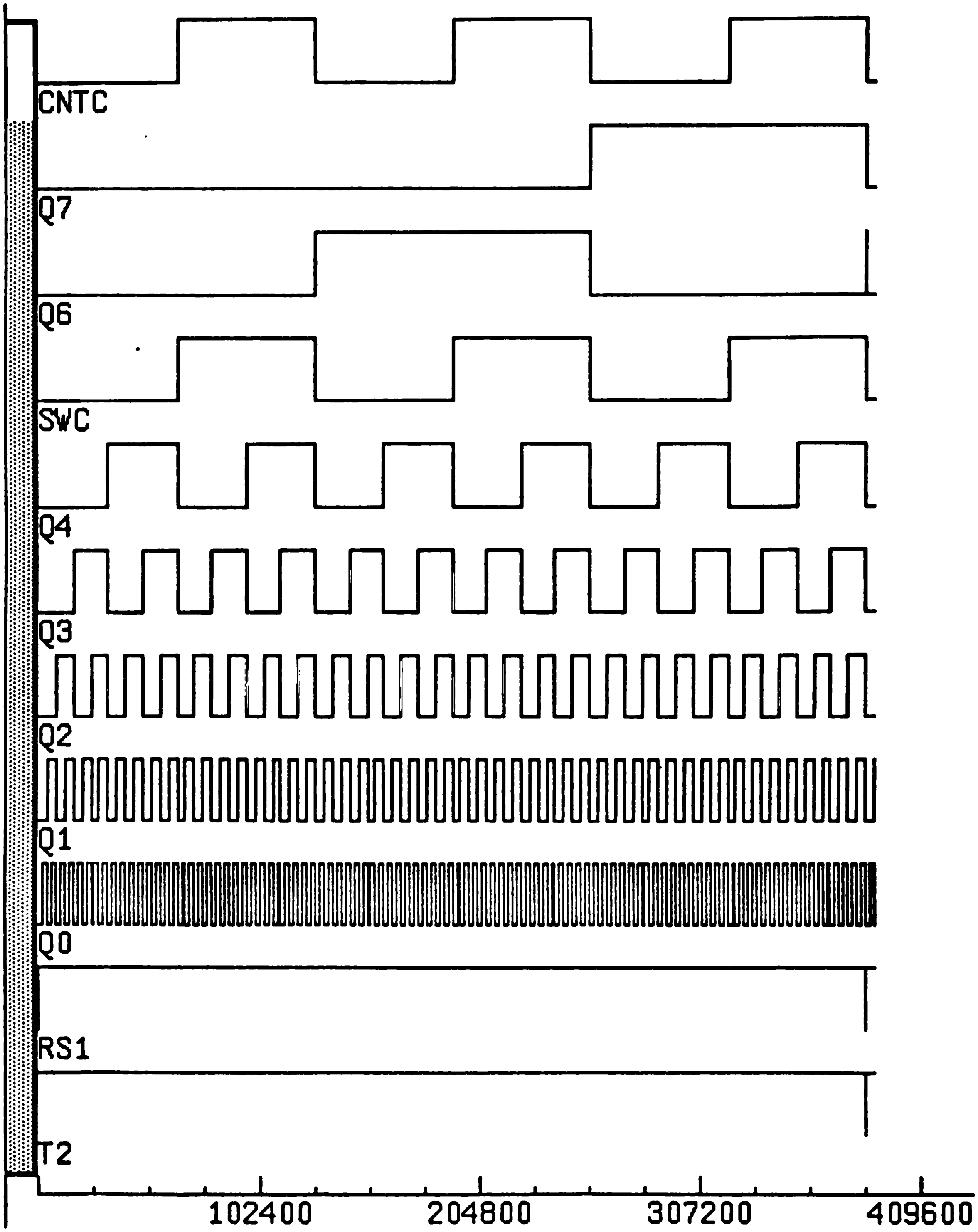


Figure 9.b - The simulation results of the CSC (counter & switch clock) generator circuit for the special case n=3, continued.

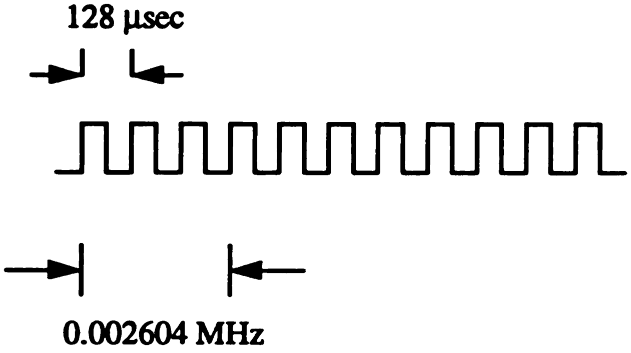


Figure 10 - The waveform SWC, with 3 pulses of 128 μsec at a 0.002604 MHz repetition frequency for the specific smart logic case.

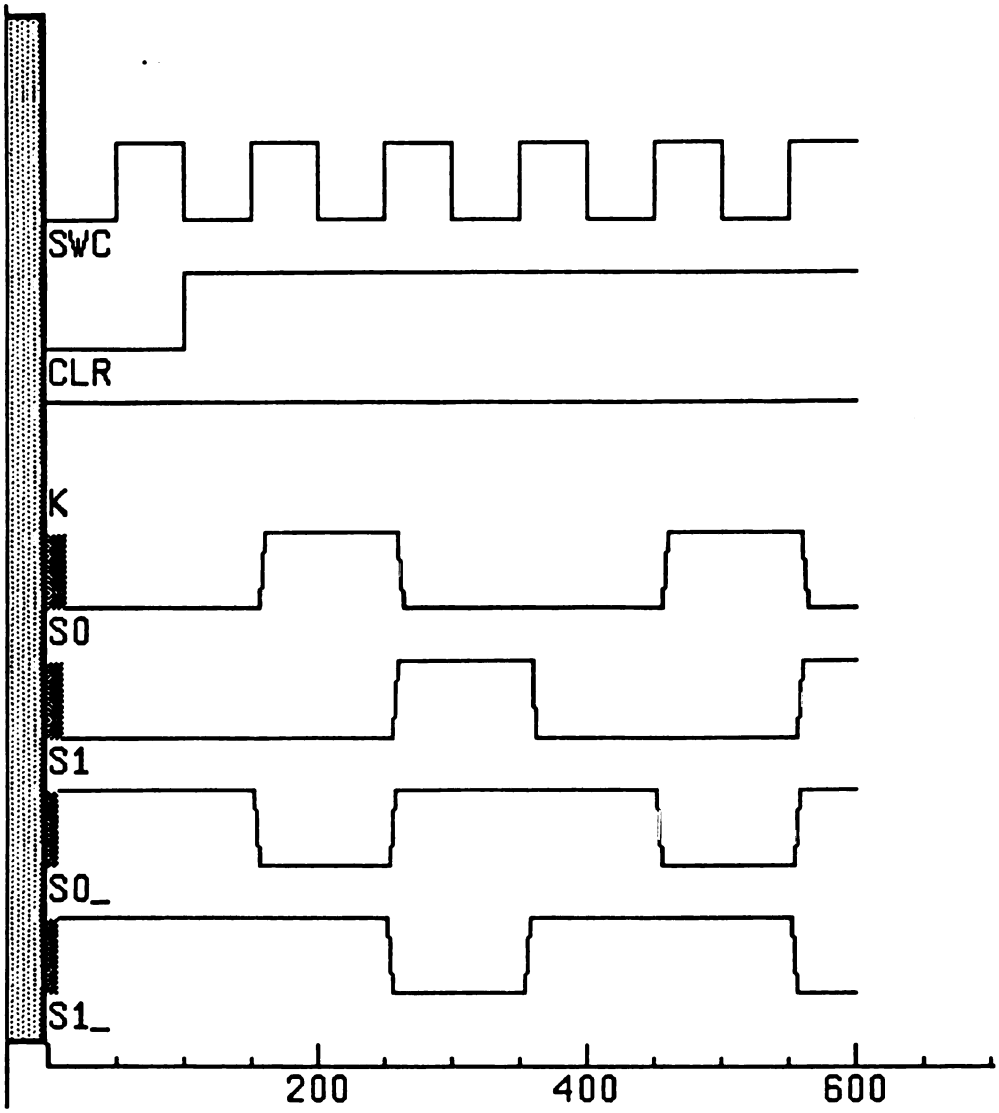


Figure 11 - The simulation results of the CTB (clock-to-binary) generator circuit for n=3

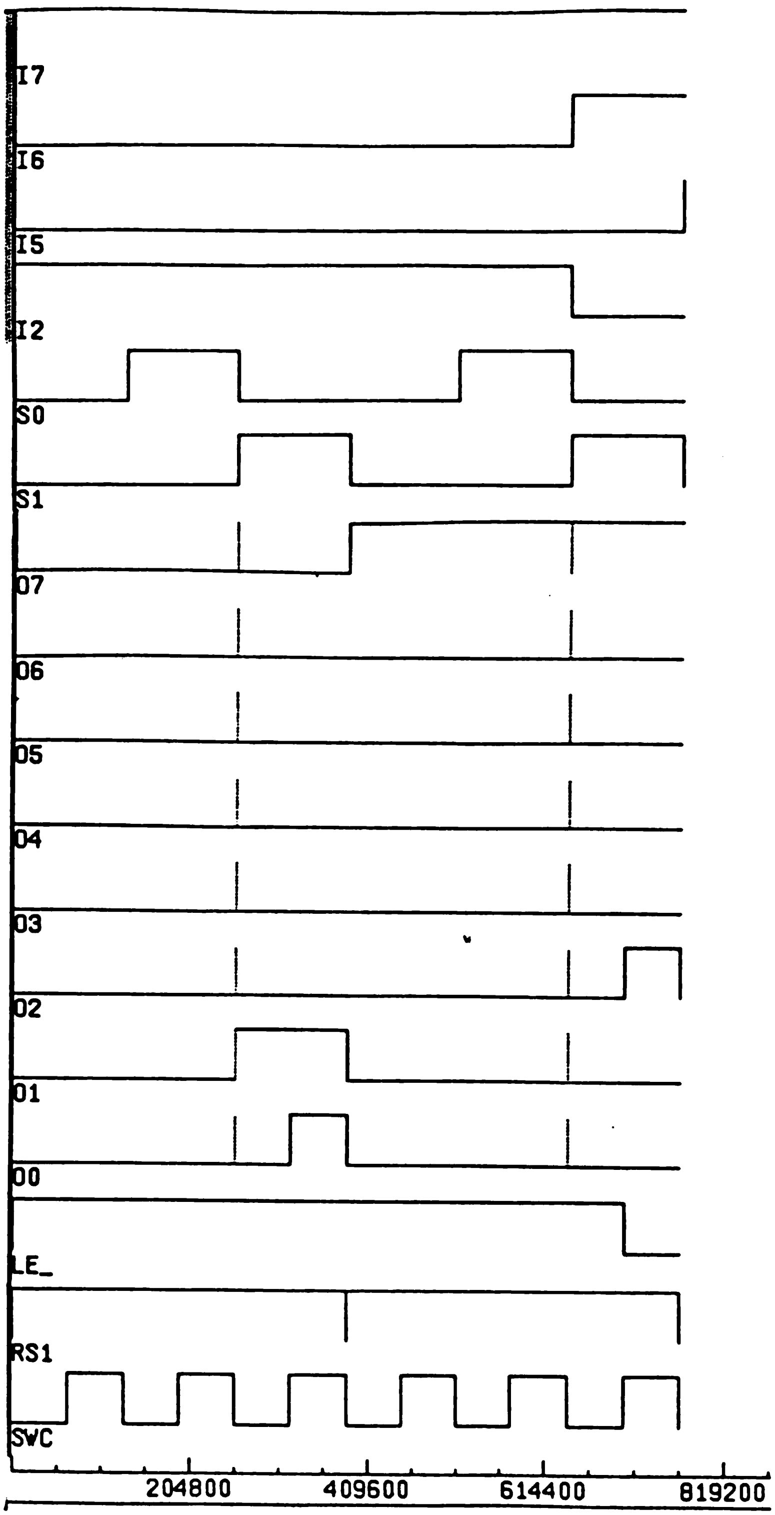


Figure 12.a - The simulation results for the latch circuit for the special case n=3, m=1, t'=125 nsec, $t_{A/D}$ =116 µsec, f=0.003 MHz.

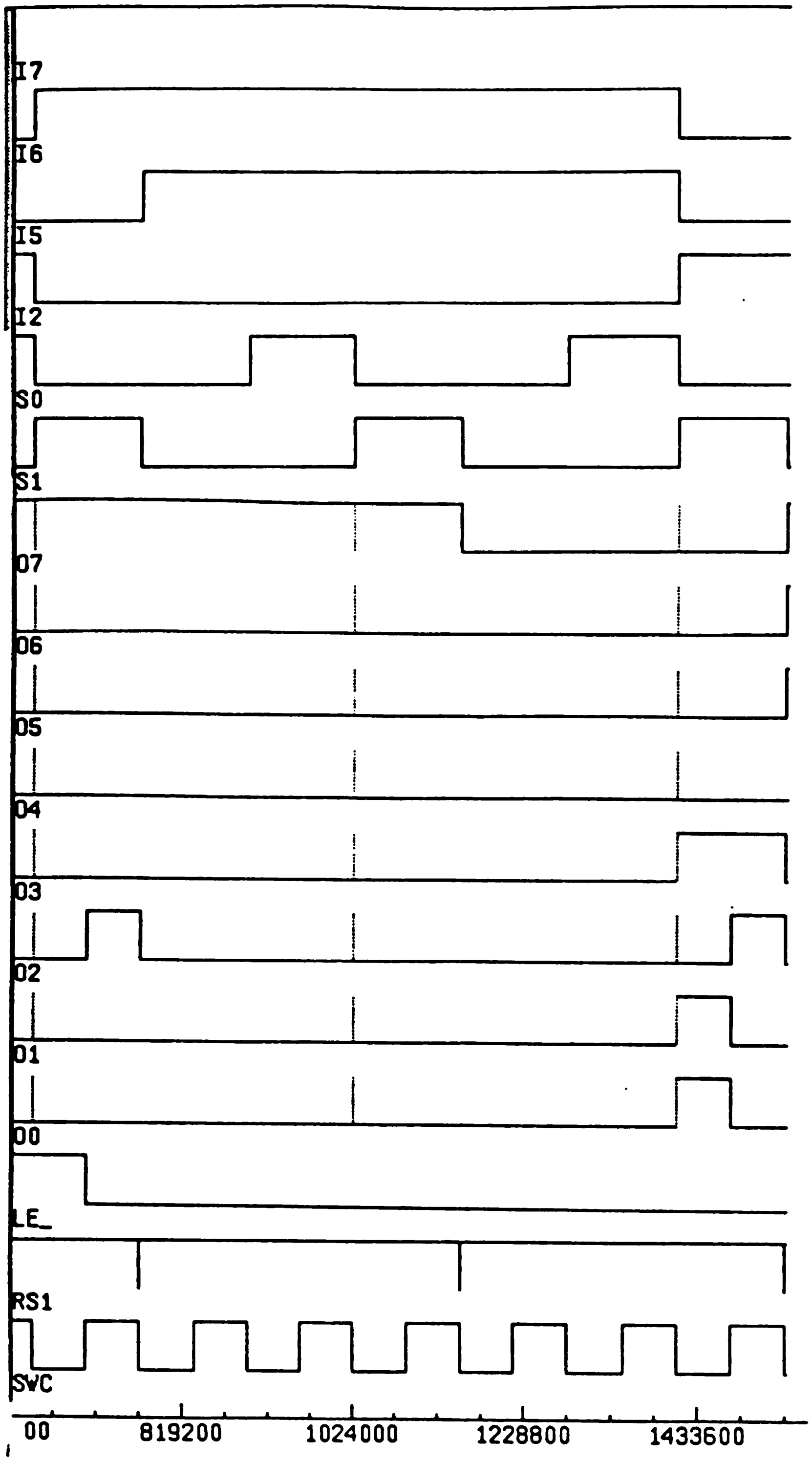


Figure 12.b - The simulation results for the latch circuit for the special case n=3, m=1, t'=125 nsec, $t_{A/D}$ =116 μsec, f=0.003 MHz, continue.

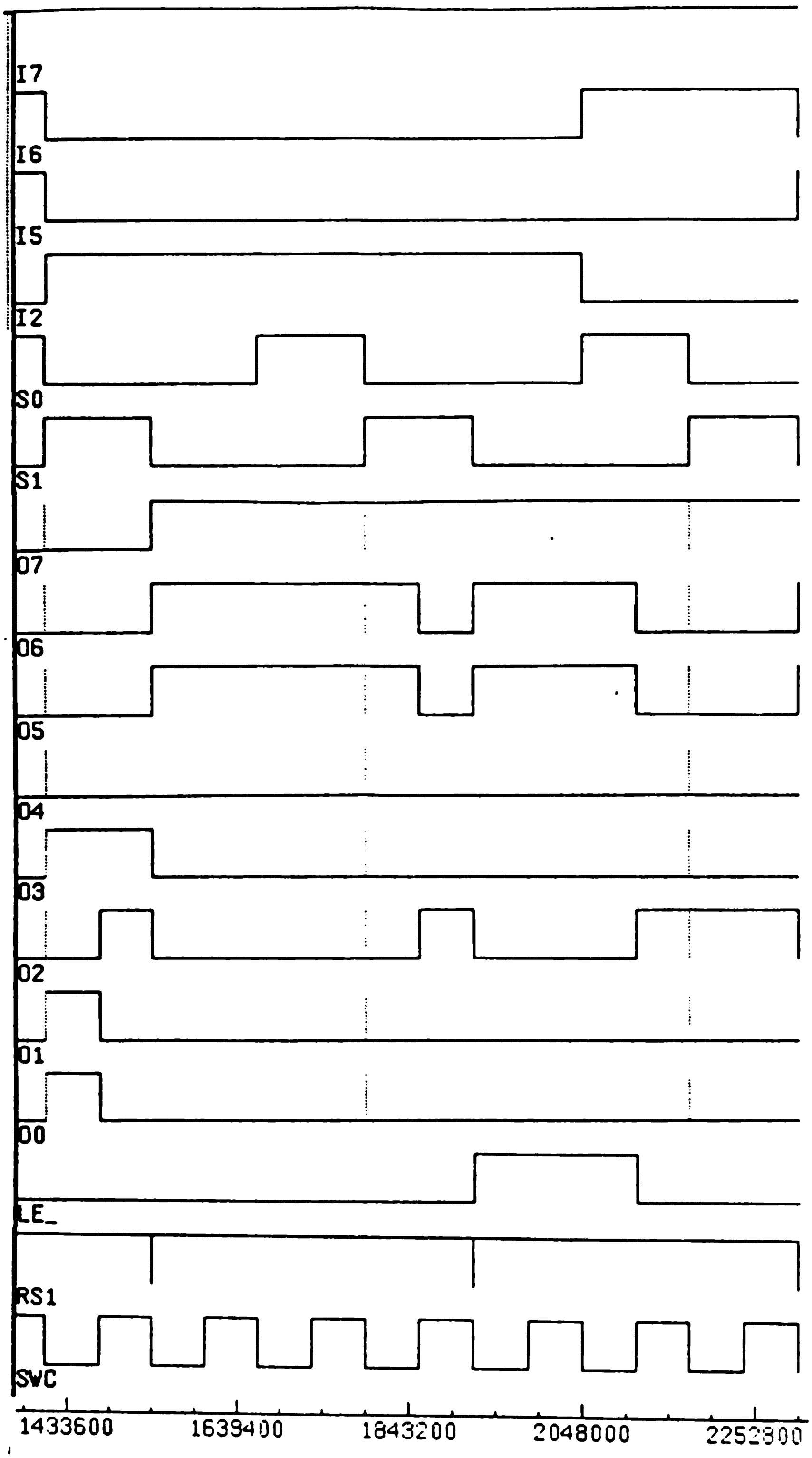


Figure 12.c - The simulation results for the latch circuit for the special case n=3, m=1, t′=125 nsec, $t_{A/D}$ =116 μsec, f=0.003 MHz, continue.

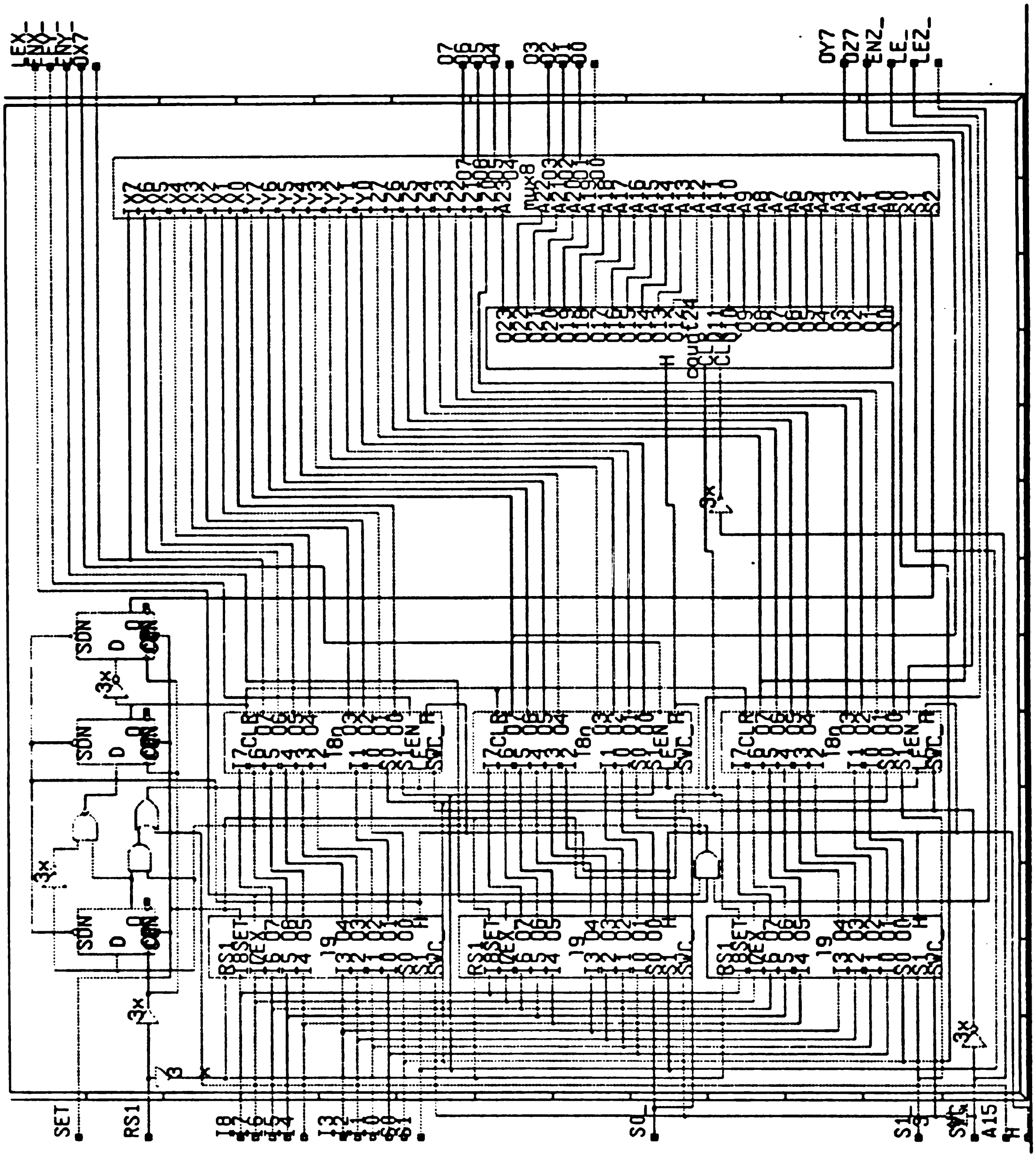


Figure 13 - The detailed circuit diagram for a two stage latch with n=3, m=1.

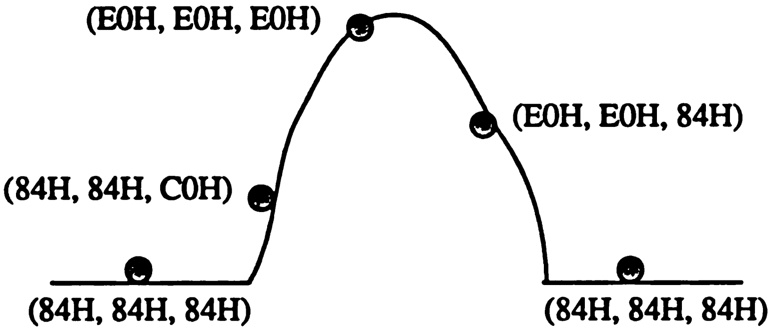


Figure 14 - One impact force which contains five sampled data.

3.5.4 COMP Design

The 8-bit comparator circuit compares the output signal from the A/D with preprogrammed thresholds (8AH and 76H, for +10 g and -10 g, respectively) and generates a latch enable signal, $LE_{i\_}$ (latch enable LE_i low true, i=x,y,z). If the signal magnitude is above the threshold, $LE_{i\_}$ will go low, otherwise, $LE_{i\_}$ will remain high.

The simulation results of the comparator circuit are shown in Figure 15. Each of the monitoring nodes can be referred to the circuit diagram shown in Figure 16. The input signals are placed on the lines A7,A6, ...,A0 (data lines) and the output signal appears as $LE_{i\_}$. The first set of input data is the sequence 00H, 10H, 20H, 30H, 40H, 50H, 60H, 70H which is above the threshold, so that $LE_{i\_}$ goes low. The second set of input data is chosen as 90H, A0H, B0H, C0H, D0H, E0H, F0H, FFH which again is above the threshold so that $LE_{i\_}$ again goes low. Finally the set of input data 76H, 78H, 7AH, 7CH, 7EH, 7FH, 80H, 81H, 82H, 84H, 86H, 88H, 8AH is applied which is below or at the threshold so that $LE_{i\_}$ goes high.

3.5.5 COUNTER Design

The 16-bit counter circuit receives the clock signal from the CSC and sends a 16-bit word to memory for address pointing. The simulation results for the counter16 circuit are shown in Figure 17. Each of the monitoring nodes can be referred to the circuit diagram shown in Figure 18. The output signals appear on the $Q_{15}Q_{14}...Q_1Q_0$ lines. The input signal is CP (CNTC). We divide the counter16 into four parts to simulate the individual circuits since simulating the whole counter16 requires 65536 clock counts which would expend to much time and paper. For each individual part the 16 clock counts are sufficient to verify counter simulation performance. Initially, RS=0 in order to clear $Q_{15}Q_{14}...Q_1Q_0$. When RS=1, $Q_{15}Q_{14}...Q_1Q_0$ will count each pulse from the clock which generates the signal CP (CNTC). This process is repeated for the other 3 four bit parts of the counter.

The block diagram for this SLIC is shown in Figure 19.

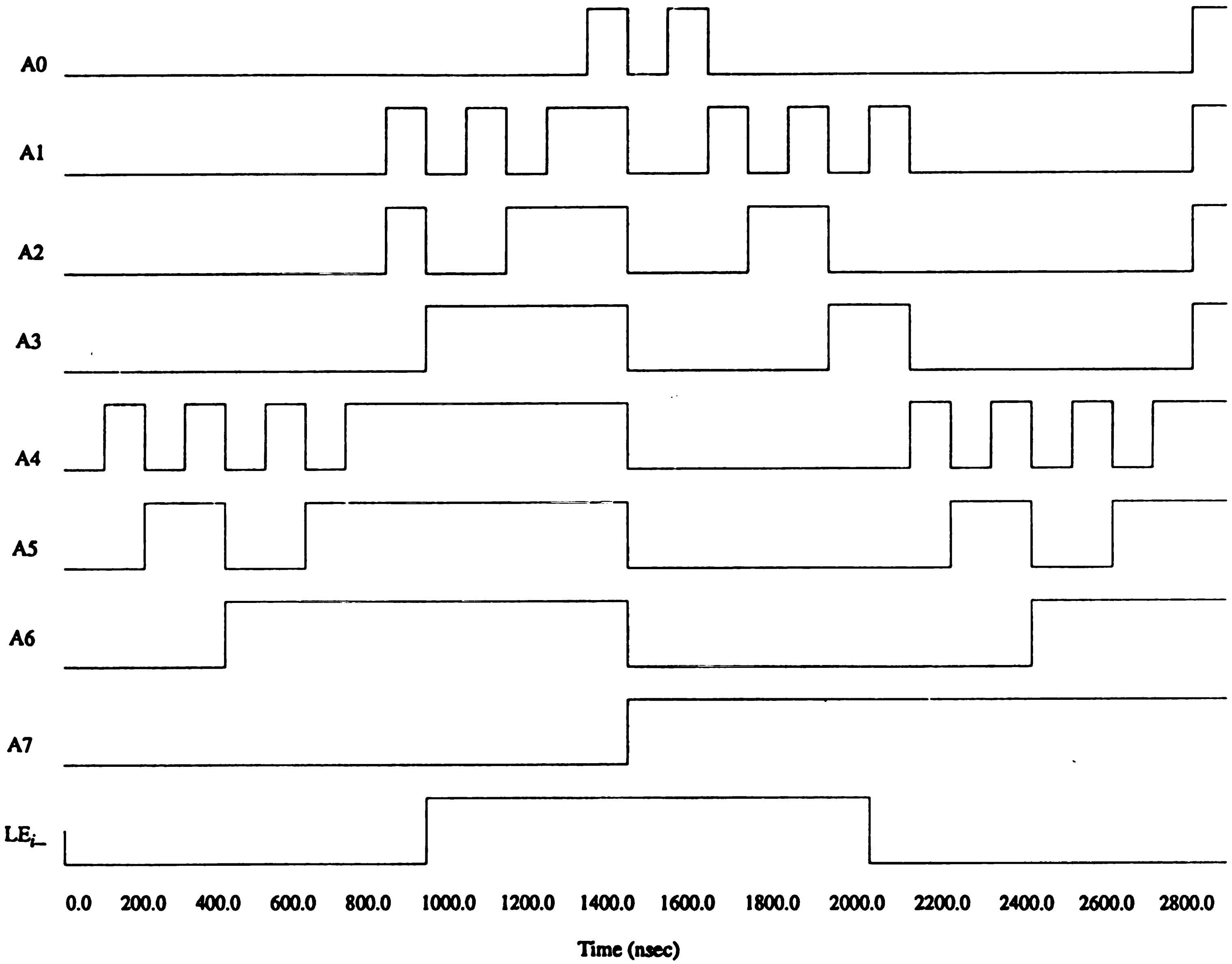


Figure 15 - The simulation results for an 8-bit comparator with predetermined thresholds.

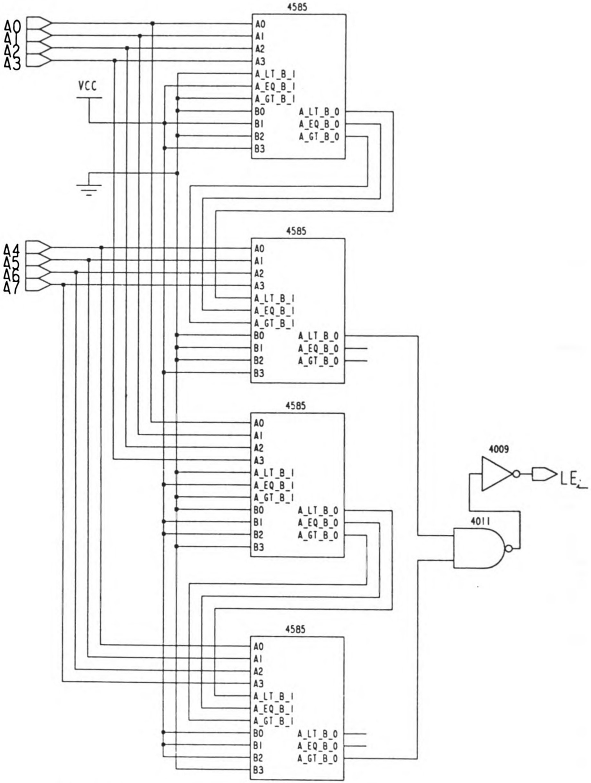
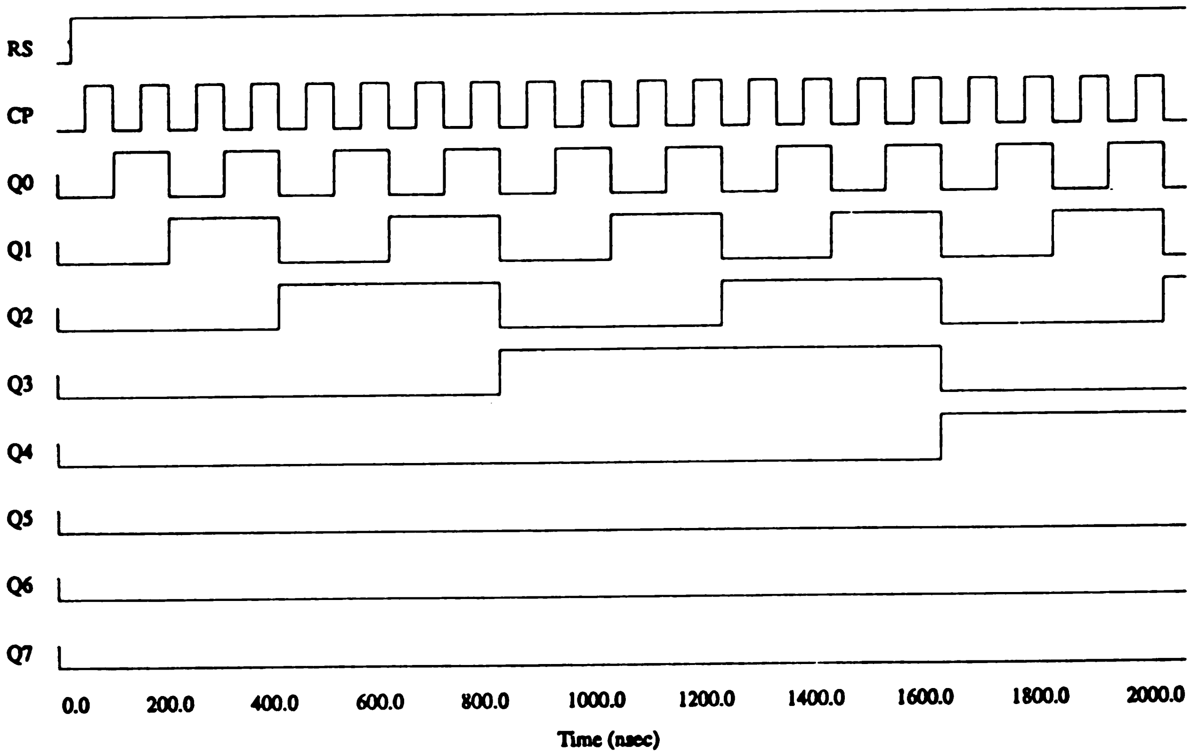


Figure 16 - The detailed circuit diagram for an 8-bit comparator with predetermined thresholds.



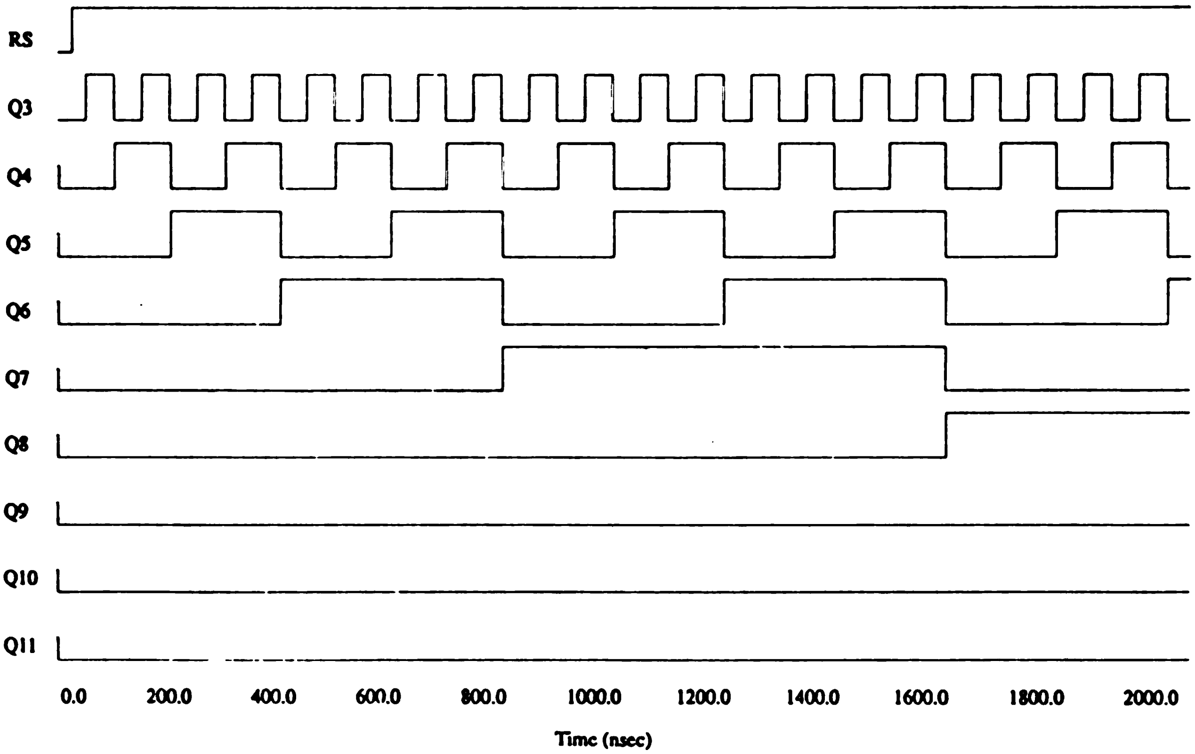
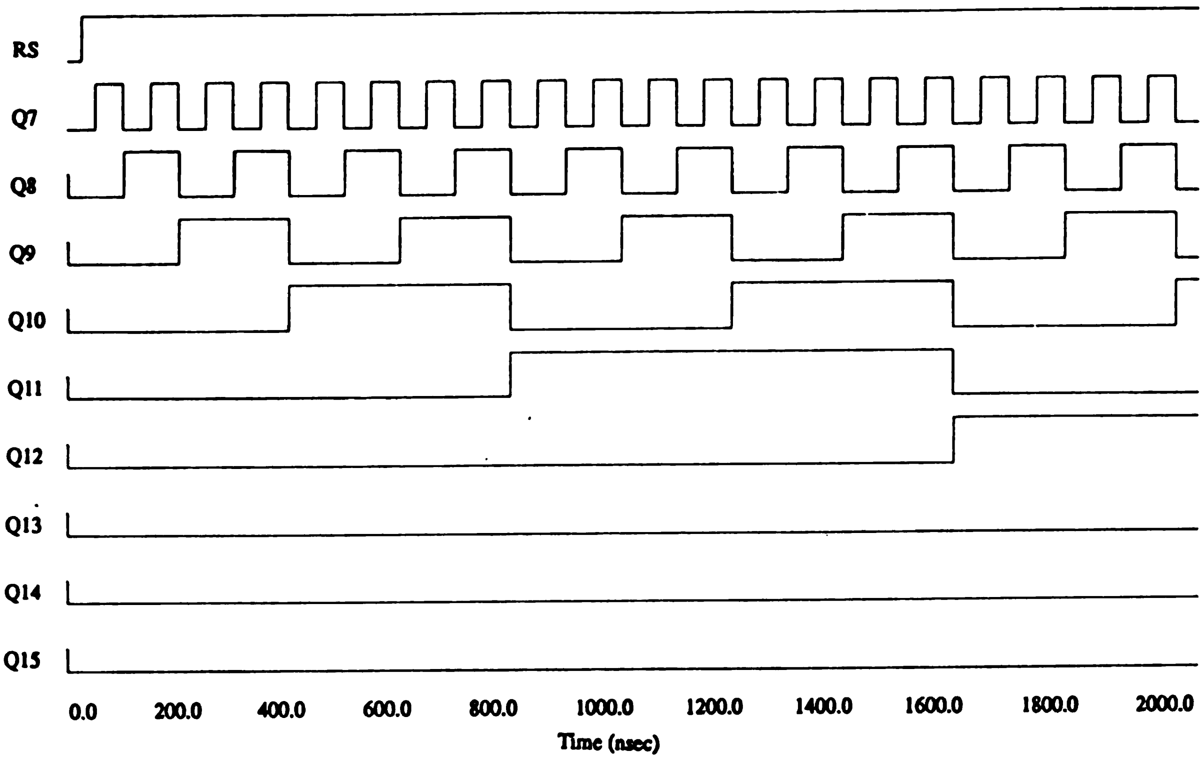


Figure 17.a - The simulation results for a 16-bit counter which is divided into four parts.



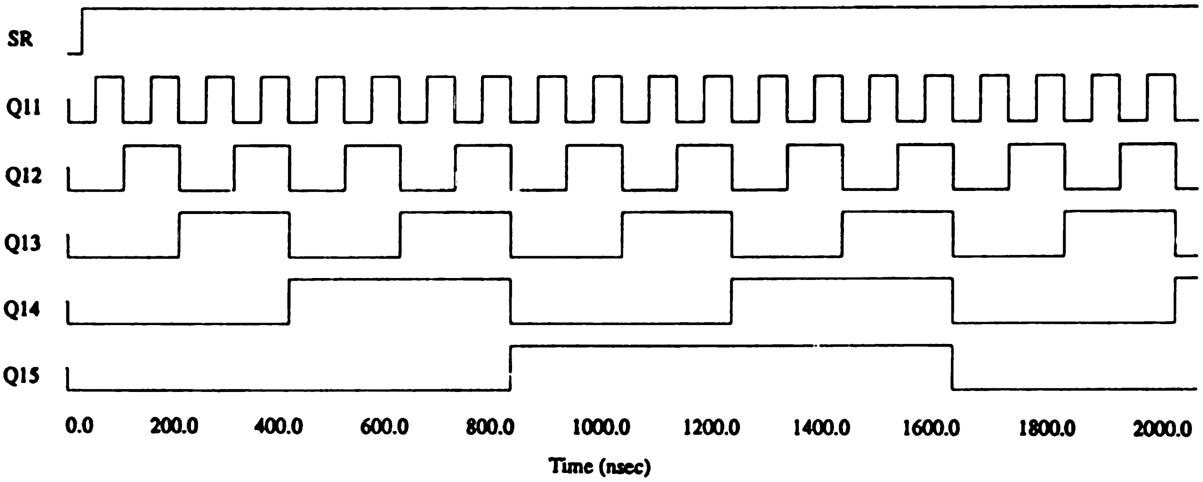


Figure 17.b - The simulation results for a 16-bit counter which is divided into four parts, continue.

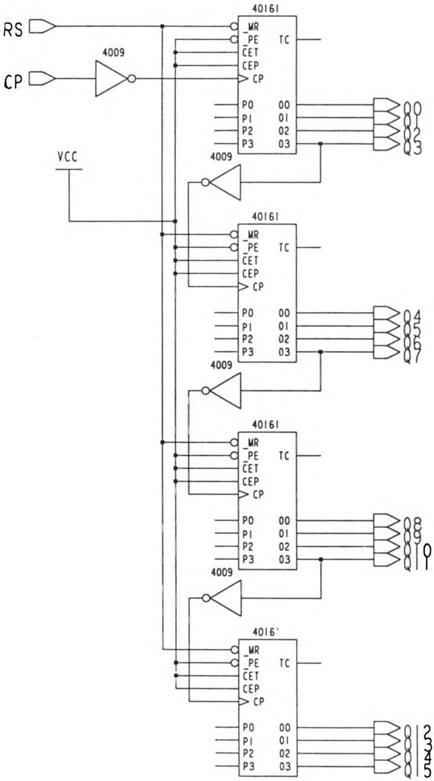


Figure 18 - The detailed circuit diagram for a 16-bit counter.

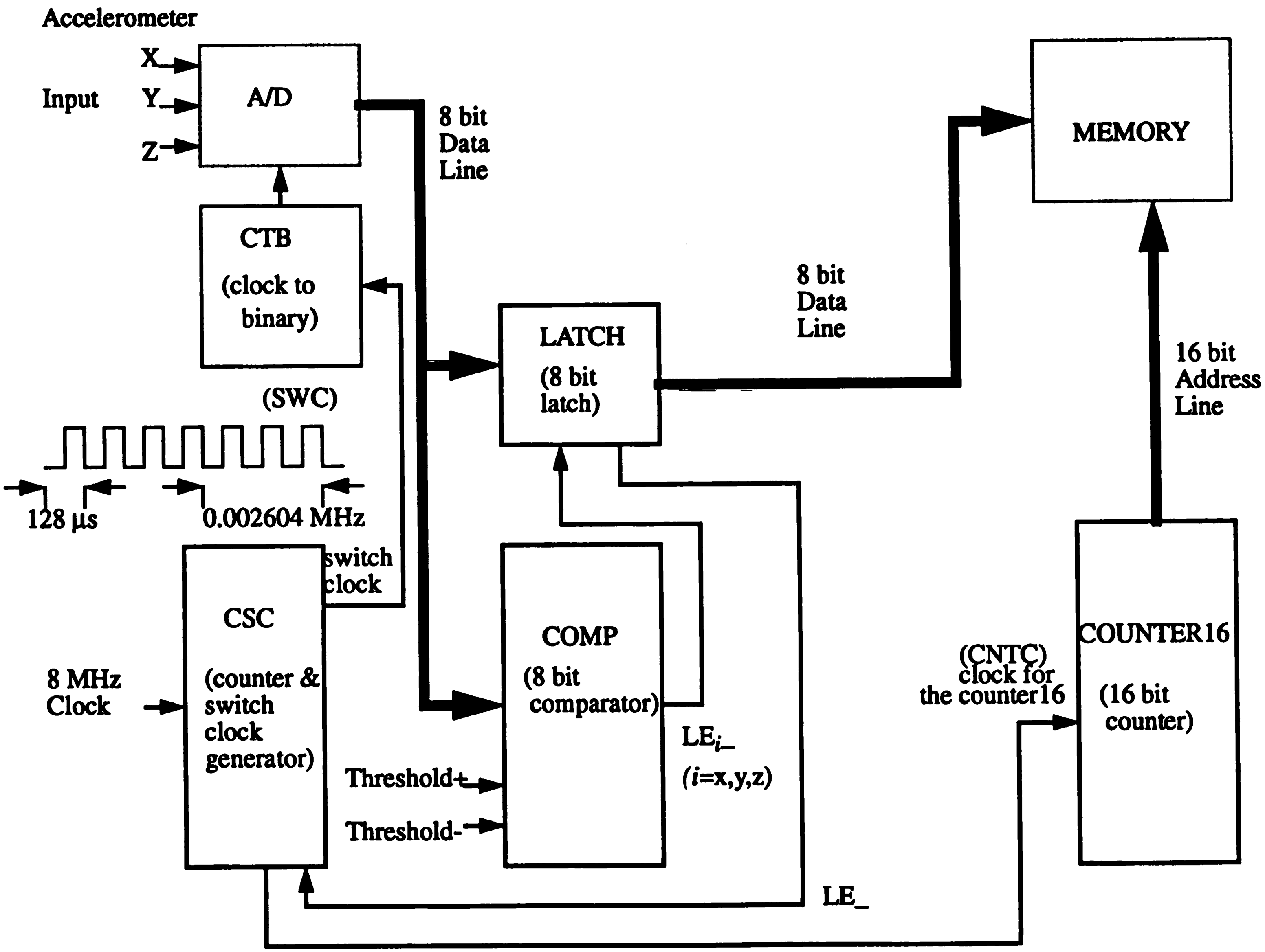


Figure 19 - Smart logic integrated circuit (SLIC) functions interfaced to memory and A/D.

3.6 Software Program

A software program based on the flow graph shown in Section 3.3 has been developed for synthesizing the n-D with m restrictions data acquisition system according to the design methodology developed in Section 3.2. The flow chart for the software to accomplish this objective is shown in Section 3.3. This program develops the actual sampling frequency $\tilde{f}$, with minimun error in the difference f- $\tilde{f}$. The detailed software listing is included in Table 1.

```
#include <math.h>
#include <stdio.h>
main()
{  /* The program for synthesizing the n-dimension with m restrictions data acquisition
system */

   float t1,t,f,tad,q1,f1,f2;
   int n,q,d,l,s,d1;
   printf("enter the t1 value with nsec:");/*internal clock period generated from the crystal*/
   t1=getnum()/1000.;
   printf("enter the f value with Hz:");  /* sampling frequency */
   f=getnum()/1000000.;
   printf("enter the tad value with µsec:");  /* A/D maximum conversion time */
   tad=getnum();
   printf("enter the n value:");  /* data dimension */
   n=getnum();
   q1=log10(tad/t1)/log10(2.0);  /* q=round-up of log2(tad/t1) */
   if (q1-(int) q1==0)
     q=q1;
   else q=++q1;
   t=t1*pow(2.0,(float) q);  /* µsec duration of the clock pulses in the burst */
   if ((1/(t*f))-(int)(1/(t*f))<0.5)  /* d=round off of 1/tf */
     d=1/(t*f);
   else d=1+1/(t*f);
   f1=(1/t)*(1/(float) d);
   l=log10((float) n)/log10(2.0)+1;  /* l=round-off of log2(n) +1 */
   d1=d&((int) pow(2.0,(float) l) -1);  /* d1 = the least l bit of d */
   if (1/(t*f)<n        /* special case: clock count per burst less than data dimension */
     f2=(1/(t*n));              /* f2 represents actual frequency of the SWC waveform */
   else if (d1==n)
        f2=(1/t)*(1/(float) d);     /* f1=f2 */
   else if (d1>n)                   /* minimize f1-f2 */
        f2=(1/t)*(1/(float) (d-d1+n));
   else if ((n-d1)<=(int) pow(2.0, (float) l-1.0))
        f2=(1/t)*(1/(float) (d-d1+n));
   else f2=(1/t)*(1/(float) (d-d1+n-(int) pow(2.0, (float) l)));
   printf("q=%d t=%f d=%d f1=%f l=%d d1=%d f2=%f", q,t,d,f1,l,d1,f2);
}
getnum()
{
   char s[80];
   gets(s);
   return(atoi(s));
}
```

Table 1 - Software program for synthesizing the n-D with m restrictions data acquisition system.

CHAPTER 4

Variation in Size, Power and Heat Dissipation with Specification Requirements

It is important to estimate the size, power and heat dissipation of the μ-P based IS, of the composite smart logic chip based IS and of the absolute limit for an optimum design in order to compare the improvement for each step of the smart logic implementation. A general analysis of power versus size for multichip housing configurations and high density VLSI will aid in the design of compact data acquisition devices.

4.1 Estimation of Size, Power and Heat Dissipation for a μ-P Based IS

The size, power and heat dissipation of the μ-P based IS are analyzed in order to compare the power and size saving achievable with a composite smart logic chip based IS.

4.1.1 Estimation of Power Dissipation in a 89 mm (3.5 in.) μ-P Based IS

A 89 mm (3.5 in.) IS previously reported by Zapp, et al. (1990) has a 13.804 mA total current consumption from a 7.2 V battery which includes 9 mA for the μ-P (the power consumption data of some typical single chip μ-P are listed in Table 2 for reference), 1.2

mA for the 32 K RAM, 0.004 mA for the latch and multiplexer, 0.6 mA for the Op Amp, and 3 mA for the regulator biased at 7.2 V. Thus the total power consumption is P=IV=13.804 mA x 7.2 V = 99.39 mW. The current and power consumption for the conventional µ-P circuit in the 89 mm (3.5 in.) IS are shown in Table 3.

According to the data sheets for the MC68HC11A8 µ-P (Motorola Semiconductor Products, Inc.), the average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J=T_A+(P_D \times \theta_{JA}) \qquad (4.1)$$

where T_A= Ambient Temperature, °C

θ_{JA} = Junction-to-Ambient Package Thermal Resistance, °C/W

$P_D = P_{INT} + P_{I/O}$

P_{INT} = Chip internal power Icc x Vcc, W

$P_{I/O}$ = Power dissipation on input and output pins

For most applications $P_{I/O} << P_{INT}$ and can be neglected.

For the 89 mm (3.5 in.) IS, the estimated µ-P current consumption based on total system current requirements gives Icc=9 mA, at Vcc=5 V, so that P_D= Icc x Vcc= 9 mA x 5 V = 45 mW if $P_{I/O}$ is neglected. Thus T_J-T_A= 45 mW x 50 °C/W (The µ-P MC68HC11A8, Plastic 52-pin Quad Pack, Thermal Resistance is given as θ_{JA}=50 °C/W) = 2.25 °C. Such a small temperature increase is allowable in the 89 mm (3.5 in.) IS.

4.1.2 Estimation of Power Dissipation in a 63.5 mm (2.5 in.) µ-P Based IS

A 63.5 mm (2.5 in.) IS previously reported by Techmark, 1991, has a 10.244 mA total current consumption which includes 9 mA for the µ-P, 1.2 mA for the 32 K RAM, 0.004 mA for the latch and multiplexer and 0.04 mA for the Op Amp at Vcc=5 V. This gives a total power consumption of P=IV=10.244 mA x 5 V = 51.22 mW.

For the 63.5 mm (2.5 in.) IS, using Eq. 4.1, and the estimated µ-P current consumption (based on total system current requirements) of Icc=9 mA, at Vcc=5 V, gives P_D= Icc x

Vcc= 9 mA x 5 V = 45 mW if $P_{I/O}$ is neglected. This results in T_J-T_A= 45 mW x 50 °C/W = 2.25 °C. This temperature increase is allowable in the 63.5 mm (2.5 in.) IS.

4.1.3 Estimation of Power Dissipation in a 50.8 mm (2.0 in.) μ-P Based IS Multichip Housing

A 50.8 mm (2.0 in.) multichip housing IS, has a 10.804 mA total current consumption which include 9 mA for the μ-P, 1.2 mA for the 32 K RAM, 0.004 mA for the latch and multiplexer and 0.6 mA for the Op Amp at Vcc=5 V, giving a total power consumption of P=IV=10.804mA x 5 V = 54.02 mW.

For the 50.8 mm (2.0 in.) IS, using Eq. 4.1, and a μ-P current consumption of Icc=9 mA, at Vcc=5 V, gives P_D= Icc x Vcc= 9 mA x 5 V = 45 mW if $P_{I/O}$ is neglected. Thus T_J-T_A= 45 mW x 50 °C/W = 2.25 °C which is allowable in the 50.8 mm (2.0 in.) IS.

Comparing section 4.1.1, 4.1.2 and 4.1.3 for the three different sizes of μ-P based IS, the T_J-T_A (the temperature difference between chip-junction and ambient) is the same because we use the same μ-P and the μ-P dominates the power consumption.

4.2 Estimation of Size, Power and Heat Dissipation for a Composite Smart Logic Chip Based IS

The SLIC, A/D, Op Amp and RAM dice are housed together on a single substrate without circuit board and connected to form what we call the composite smart logic chip. Figure 20 depicts the smart logic substrate layout, showing the 4 subsets, RAM, Op Amp, SLIC and A/D. The dimensions of the single housing unit are 20 x 20 mm (0.787 x 0.787 in.). The composite smart logic chip will save power compared to the previous 89 mm (3.5 in.) IS since SLIC replaces the μ-P which dominates the total power consumption. It is necessary to estimate the size and power for a composite smart logic chip based IS to verify the specification requirements of smaller size and lower power consumption than for the previous 89 mm (3.5 in.) IS.

4.2.1 Estimation of the Size

Three different configurations for the smart logic chip based IS are considered:

For a one substrate composite smart logic chip configuration, the volume of the substrate is 20 mm x 20 mm x 5 mm (0.787 in. x 0.787 in. x 0.197 in.), the volume of the triaxial piezoelectric accelerometer (Vibrametrics model 3130HT) is 7 mm x 7 mm x 15.2 mm (0.275 in. x 0.275 in. x 0.6 in.), and the volume of the smallest battery (Panasonic model PZ2330) of which 3 are needed has a 23.2 mm diameter and a 3.0 mm height (0.91 in. diameter, 0.12 in. height). Using Autocad an optimal volume configuration has been developed as shown in Figure 21. The components are contained in a spherical volume which has a sphere diameter of 30 mm (1.18 in.). If a utilization rate (minimum element volume divided by minimum sphere space) of 46.33% is assumed and the remainder of the sphere is filled with heat sink, rigid foam, beeswax, and epoxy walls of 3.81 mm (0.15 in.) thickness, the IS diameter becomes 37.62 mm (1.48 in.). The disadvantage of this configuration is that the accelerometer is not located in the center of the sphere.

For two half-substrate configurations, the volume of each substrate is 20 mm x 10 mm x 5 mm (0.787 in. x 0.394 in. x 0.197 in.). Using the same accelerometer and battery as previously, the Autocad software provides two possible configurations. The sphere diameter for one is 31.62 mm (1.25 in.) which for a utilization rate of 39.57% gives an IS diameter of 39.24 mm (1.55 in.) as shown in Figure 22. The advantage of this configuration is that the accelerometer is located near the center of the sphere.

The sphere diameter for the third configuration is 30.24 mm (1.19 in.) which using a utilization rate of 45.24% gives an IS diameter of 37.86 mm (1.49 in) as shown in Figure 23. The advantage of this configuration is that the accelerometer is located in the sphere center. The disadvantage is that the separating distance between the two half-substrates is larger than that of the previous configuration, so that longer wire connections are necessary.

The detailed IS dimensions for the three different configurations are shown in Table 4.

4.2.2 Estimation of the Power and Heat Dissipation

Replacement of the μ-P with a composite smart logic chip results in a substantial power reduction. The expected power reduction can be calculated using the known power consumption of individual gates comprising the composite smart logic chip. Specifically, the gate count for the SLIC is approximately 1500 gates. According to the National Semiconductor CMOS Logic Databook, Application Note 303, Kenneth Karakotsios [54], the CMOS Logic Databook [55], the Texas Instruments CMOS Logic Databook [56], and the Samsung CMOS Logic Databook [57], as summarized in Table 5, the average HCMOS current consumption for one gate is 0.5 μA at T_A=25 °C, at a bias voltage of Vcc= 6 V. Thus the current consumption of the smart logic circuit is less than 1 mA at Vcc= 5 V. The current consumption of the 32 K RAM (IDT 71256, Integrated Device Technology, Inc., Santa Clara, CA) is approximately 1.2 mA at Vcc= 5 V. The current consumption of the A/D (ADC 0808/ADC 0809, National Semiconductor) is approximately 3 mA at Vcc= 5 V. The current consumption of the Op Amp (TLC 27L4, Texas Instruments Inc., Dallas, TX) is approximately 0.04 mA at Vcc= 5 V (A recent modification incorporated in the IS by Techmark Inc. of Lansing uses this type of Op Amp to provide a reduced current consumption of 0.04 mA and achieve a smaller size for the IS of 63.5 mm (2.5 in.) diameter.), so that the total current consumption of the composite smart logic device during operation is about 5.24 mA at Vcc= 5 V. This is less than half the current consumption of the conventional μ-P based circuit. The better than 2.0 reduction in power consumption allows the use of smaller volume batteries to achieve the same results as for the existing larger IS. The current and power consumption for the composite smart logic IS chip are shown in Table 3.

According to R. C. Eden [58], the maximum temperature rise, T_m, above the heat sink temperature is given by:

$$T_m-T_o= (P/2\pi kz)[\ln(R_o/R_i)+1/2] \qquad (4.2)$$

where k is the substrate thermal conductivity, z its thickness, T_o is the heat sink temperature, P is the power which is uniformly distributed over the chip area, and R_i is the radius of a chip centered on a substrate of radius R_o.

For the composite smart logic chip (38 mm or 1.5 in. IS), with a silicon substrate of thickness z=0.5 mm, thermal conductivity of k = 149 W/m°C [58], substrate of diameter $2R_o$=28.4 mm (1.12 in.) (actual chip is 19 x 19 mm (0.75 x 0.75 in.) square) and four dice distributed in a diameter of $2R_i$=14.2 mm (0.56 in.) gives a power consumption of P= 5.24 mA x 5 V =26.2 mW. Using Eq. 4.2,

$$T_m-T_o=[26.2\text{ mW}/(2\pi \times 149\text{ W/m}^\circ\text{C} \times 0.5\text{ mm})]\times[\ln(28.4\text{ mm}/14.2\text{ mm})+1/2]=0.067\ ^\circ\text{C}$$

which is smaller than the temperature increase for the 89 mm (3.5 in.) IS. Thus power dissipation represents no problem for size reduction related to multichip housing.

4.3 Ultimate Size, Power and Heat Sink Limitations

Assuming the availability of smaller accelerometers and batteries, three different size limitations are considered:

For a one full-substrate configuration, using a smaller battery with dimensions of 16.64 mm diameter and 3.0 mm height (0.655 in. diameter, 0.12 in. height), or approximately 51% of the original battery, we can get the lowest bound on the sphere diameter of 28.72 mm (1.13 in.) as shown in Figure 24. With the addition of rigid foam, beeswax, and epoxy shell, the IS diameter becomes 36.34 mm (1.43 in.).

For a two half-substrate configuration, using a battery with dimensions of 13.4 mm diameter and 2.5 mm height (0.528 in. diameter, 0.1 in. height), or 18 % the volume of the original battery and a smaller accelerometer of dimensions 5 mm x 5mm x 12.43 mm (0.197 in. x 0.197 in. x 0.49 in.), we get the lowest bound on the sphere diameter of 24.58 mm (0.97 in.) as shown in Figure 25. Adding the rigid foam, beeswax, and epoxy shell increases the IS diameter to 33.1 mm (1.27 in.). For this configuration, the accelerometer would not be located exactly in the sphere center. To locate the accelerometer in the sphere center requires two substrates and longer wire connections thus increasing the dimensions slightly to 28.28 mm (1.12 in.) for the basic unit as shown in Figure 26 and to 35.9 mm (1.42 in.) for the rigid foam, beeswax and epoxy shell encased unit. For the minimum size in this latter configuration, the battery dimensions were chosen to be 19.66 mm diameter and 2.5 mm height (0.744 in. diameter, 0.1 in. height).

The detailed limited IS dimensions for the three different configurations are shown in Table 6.

To achieve an even smaller size IS (< 33 mm or 1.3 in.), only the size of the accelerometer and batteries must decrease. The power consumption and substrate size will remain the same as for the 38 mm (1.5 in.) smart logic based IS. The temperature increase is thus the same as for the 38 mm (1.5 in.) IS if the same silicon substrate be used. This certainly will cause no thermal concern. However, if it is desirable to prevent even such a small temperature increase, a high thermal conductivity substrate could be chosen. Diamond's physical attributes make it a very good substrate material choice [58]~[61]. Diamond provides a combination of good heat conduction and excellent electrical insulation with a low dielectric constant. For example, the thermal conductivity of CVD diamond = 1000 ~ 1300 W/m°C (Lateral, in-plane), 1250-1600 W/m°C (Normal, Top-bottom) for ET-100 quoted by Norton Company; and > 1300 W/m°C (Lateral), > 1600 W/m°C (Normal) for ET-200 again from Norton Company. These values are 2 to 4 times superior to that of copper. Of course, the structural strength of diamond is also highly desirable in impact measuring devices such as the IS.

The freestanding "white" diamond wafer, the purest form of diamond manufactured by the chemical vapor deposition (CVD) process, is now available from Norton Company's diamond film division with dimensions of 100 mm (4 in.) diameter and 1 mm (0.04 in.) thickness. This achievement represents an important breakthrough in the commercialization of diamond produced by the CVD process, which can be utilized directly in the IS. If a diamond substrate replaces the silicon substrate in the minimum sized IS, according to Eq. 4-2, with k=1000 W/m°C, and the other parameters the same as in Section 4-2, the 38 mm (1.5 in.) smart logic based IS gives a temperature rise of :

T_m-T_o=[26.2 mW/(2π x 1000 W/m°C x 0.5 mm)]x[ln(28.4 mm/14.2 mm)+1/2]=0.01 °C which is one sixth the temperature increase of the silicon substrate. This small temperature increase is clearly acceptable for the minimum sized IS.

Many IC now operate from 3 V power supplies [62],[63]. Embedded controller, memories, and a variety of logic chips are widely available in this form. Choice of the low voltage ICs will allow the battery requirements to decrease (the three cells may be reduced to two cells) and the battery to last longer between recharges. Other benefits include smaller size and lower heat dissipation than for 5 V systems.

4.4 Variation of Power and Size vs Specification Change

The estimation of power and size has been accomplished in section 4.2 for the specified parameter values for the IS, namely, t′=0.125 µsec, $t_{A/D}$=116 µsec, f=0.003 MHz and n=3. If the same size and power algorithm is applied to other systems, the parameter values may be different, but the variation in power and size will be small. The gate count for the smart logic circuit dominates the power and size constraint if the remaining circuit elements are maintained. An analysis of the variation of gate count vs specification change is provided. A gate count corresponds to 2.5 µW in power and approximately 0.003 mm^2 (4.65 x 10^{-6} $in.^2$) in area.

Assuming fixed values for t′, f, and n as specified above and allowing for a variation in $t_{A/D}$ of 2 µsec, 4 µsec, 8 µsec, 16 µsec, 32 µsec, 64 µsec and 116 µsec, we obtain different 1/tf and f″ which produce the gate count variation shown in Table 7.a and Figures 27 & 28. From this table, the gate count variation vs different specifications relative to the $t_{A/D}$ =116 µsec is one gate count, or 2.5 µW in power change (referred to Table 8) and no change in area (referred to Appendix 2 & 3).

In similar fashion, if we alter f from 0.003 MHz to 0.0025 MHz keeping the remainding variables fixed and following the same procedure as above, the result are as shown in Table 7.b and Figures 27 & 28. From this table, the gate count variation vs different specifications relative to $t_{A/D}$=116 μsec is at most one gate count, or 2.5 μW in power (referred to Table 8) and no change in area (referred to Appendix.2 & 3).

Manufacturer	Model No.	Max supply current	ROM	RAM	EEPROM
Motorola	MC68HC11A8	20 mA	8 K bytes	256 bytes	512 bytes
Intel	8050AH	80 mA	4 K bytes	256 bytes	
	8049AH	70 mA	2 K bytes	128 bytes	
	8048AH	65 mA	1 K bytes	64 bytes	
	8052AH	175 mA	8 K bytes	256 bytes	
	8051AH	125 mA	4 K bytes	128 bytes	

Table 2 - Maximun μ-P current consumption for different models from two manufacturers.

	conventional μ-P circuit (89 mm or 3.5 in. IS)	composite smart logic chip (38 mm or 1.5 in. IS)
μ-P	9.0 mA	
Memory (32 K)	1.2 mA	1.2mA
MUX	0.002 mA	
Latch	0.002 mA	
Op Amp	0.6 mA	0.04 mA
Regulator	3.0 mA	
Smart Logic Circuit		1.0 mA
A/D		3.0 mA
Total current consumption	13.804 mA	5.24 mA
Total power consumption	99.39 mW	26.2 mW

Table 3 - Current and power consumption for the conventional μ-P circuit (89 mm or 3.5 in. IS) and the composite smart logic chip.

sub

mi

ele

co

uti

substrate type	one full-substrate	two half-substrates	two half-substrates
minimum diameter	30.00 mm	31.62 mm	30.24 mm
	(1.18 in.)	(1.25 in.)	(1.19 in.)
element volume	6.55 cm^3	6.55 cm^3	6.55 cm^3
	(0.4 in.^3)	(0.4 in.^3)	(0.4 in.^3)
complete sphere space	14.14 cm^3	16.55 cm^3	14.48 cm^3
	(0.86 in.^3)	(1.01 in.^3)	(0.88 in.^3)
utilization rate	46.33%	39.57%	45.24%

Table 4 - IS dimensions for different geometrical configurations.

Manufacturer	Gate Type	T_A	Icc	Vcc
National Semiconductor	Quad Gates	25 °C	2.0 µA	6 V
	Quad Nand Gate	25 °C	2.0 µA	6 V
	Quad Nor Gate	25 °C	2.0 µA	6 V
Texas Instruments	Quad Nand Gate	25 °C	2.0 µA	6 V
	Quad Nor Gate	25 °C	2.0 µA	6 V
Samsung	Quad Nand Gate	25 °C	2.0 µA	5 V
	Quad Nor Gate	25 °C	2.0 µA	5 V

Table 5 - CMOS current consumption values for various quad gates at T_A=25 °C for different manufacturers.

substrate type	one full-substrate	two half-substrates	two half-substrates
minimum diameter	28.72 mm	24.58 mm	28.28 mm
	(1.13 in.)	(0.97 in.)	(1.12 in.)
element volume	4.70 cm^3	3.02 cm^3	4.20 cm^3
	(0.29 $in.^3$)	(0.18 $in.^3$)	(0.26 $in.^3$)
complete sphere space	12.40 cm^3	7.78 cm^3	11.84 cm^3
	(0.76 $in.^3$)	(0.47 $in.^3$)	(0.72 $in.^3$)
utilization rate	37.91%	38.78%	35.50%

Table 6 - Minimum IS dimensions for different geometrical configurations.

$t_{A/D}$	t'	f	n	1/tf		$1/tf^{\sim}$	$f^{\sim}$	gate count variation
μsec	μsec	MHz		decimal	binary	binary	MHz	
2	0.125	0.003	3	167	10100111	10100111	0.002994	1
4	0.125	0.003	3	83	01010011	01010011	0.003012	1
8	0.125	0.003	3	42	00101010	00101011	0.002907	1
16	0.125	0.003	3	21	00010101	00010111	0.002717	1
32	0.125	0.003	3	10	00001010	00001011	0.002841	1
64	0.125	0.003	3	5	00000101	00000111	0.002232	1
116	0.125	0.003	3	3	00000011	00000011	0.002604	0

Table 7.a - Gate count variation vs different specifications relative to $t_{A/D}$ =116 μsec and maintaining t'=0.125 μsec, f=0.003 MHz, and n=3 conditions.

$t_{A/D}$	t'	f	n	1/tf		$1/tf^{\sim}$	$f^{\sim}$	gate count variation
µsec	µsec	MHz		decimal	binary	binary	MHz	
2	0.125	0.0025	3	200	11001000	11000111	0.002513	1
4	0.125	0.0025	3	100	01100100	01100011	0.002525	1
8	0.125	0.0025	3	50	00110010	00110011	0.002451	1
16	0.125	0.0025	3	25	00011001	00011011	0.002315	1
32	0.125	0.0025	3	13	00001101	00001111	0.002083	1
64	0.125	0.0025	3	6	00000110	00000111	0.002232	1
116	0.125	0.0025	3	3	00000011	00000011	0.002604	0

Table 7.b - Gate count variation vs different specifications relative to $t_{A/D}$ =116 µsec keeping t'=0.125 µsec, f=0.0025 MHz, and n=3 fixed.

$t_{A/D}$	t'	f	n	$1/tf$		$1/tf^{\sim}$	$f^{\sim}$	power variation	
μsec	μsec	MHz		decimal	binary	binary	MHz	μA	μW
2	0.125	0.003	3	167	10100111	10100111	0.002994	0.5	2.5
4	0.125	0.003	3	83	01010011	01010011	0.003012	0.5	2.5
8	0.125	0.003	3	42	00101010	00101011	0.002907	0.5	2.5
16	0.125	0.003	3	21	00010101	00010111	0.002717	0.5	2.5
32	0.125	0.003	3	10	00001010	00001011	0.002841	0.5	2.5
64	0.125	0.003	3	5	00000101	00000111	0.002232	0.5	2.5
116	0.125	0.003	3	3	00000011	00000011	0.002604	0	0
2	0.125	0.0025	3	200	11001000	11000111	0.002513	0.5	2.5
4	0.125	0.0025	3	100	01100100	01100011	0.002525	0.5	2.5
8	0.125	0.0025	3	50	00110010	00110011	0.002451	0.5	2.5
16	0.125	0.0025	3	25	00011001	00011011	0.002315	0.5	2.5
32	0.125	0.0025	3	13	00001101	00001111	0.002083	0.5	2.5
64	0.125	0.0025	3	6	00000110	00000111	0.002232	0.5	2.5
116	0.125	0.0025	3	3	00000011	00000011	0.002604	0	0

Table 8 - Power variation vs different specifications relative to $t_{A/D}$ =116 μsec maintaining t'=0.125 μsec, f=0.003 MHz, and n=3 fixed.

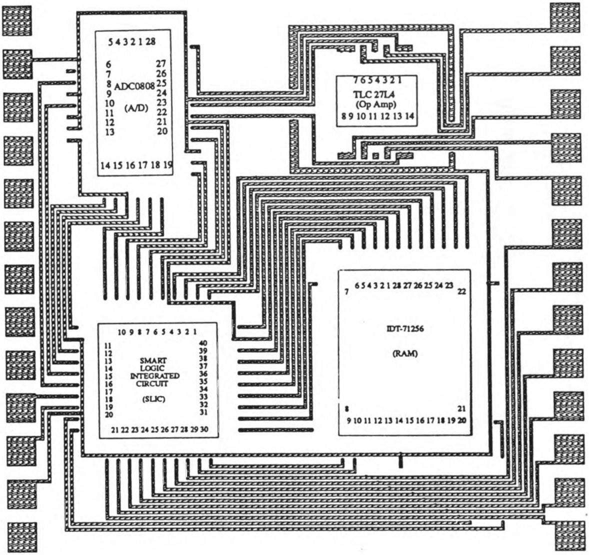


Figure 20 - Patterned silicon substrate to support the SLIC, RAM, A/D and Op Amp dice in order to form the composite smart logic chip.

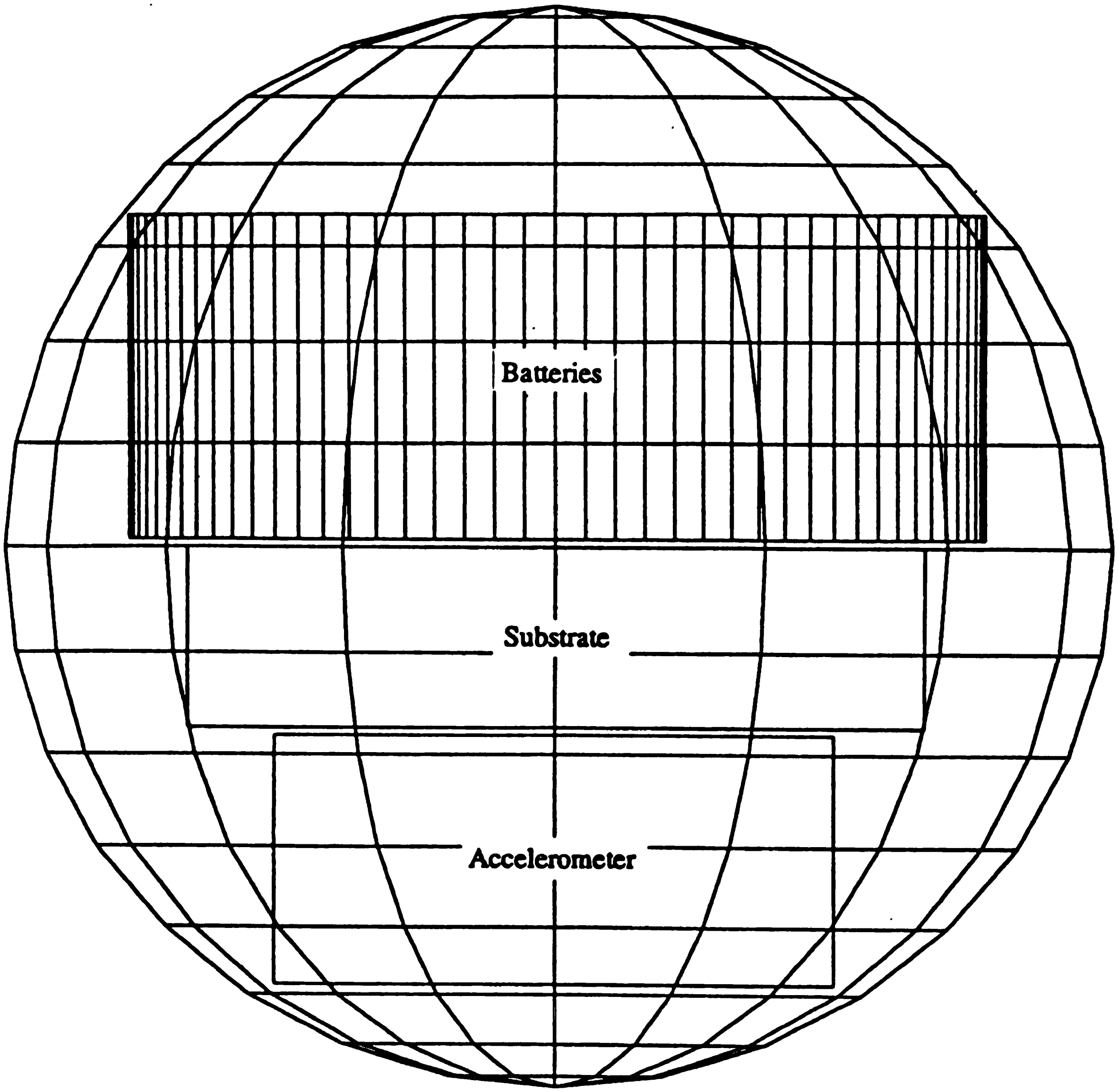


Figure 21 - The front view of the typical configuration for the sphere of 30 mm (1.18 in.) diameter. Not included: rigid foam, beeswax, and epoxy.

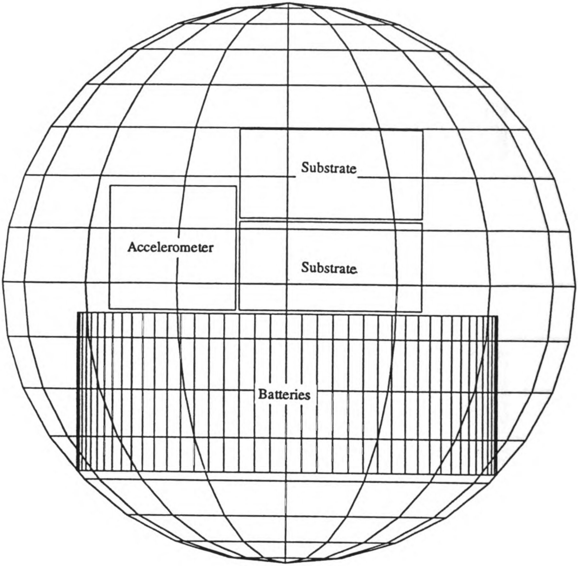


Figure 22 - The side view of the typical configuration for the sphere of 31.62 mm (1.25 in.) diameter. Not included: rigid foam, beeswax, and epoxy.

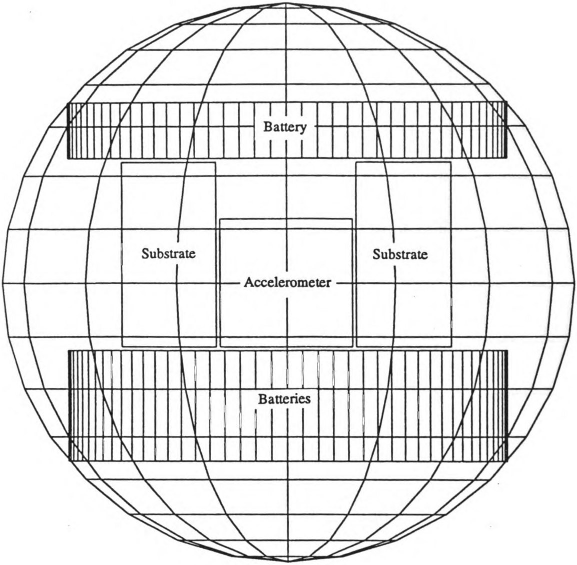


Figure 23 - The front view of the typical configuration for the sphere of 30.24 mm (1.19 in.) diameter. Not included: rigid foam, beeswax, and epoxy.

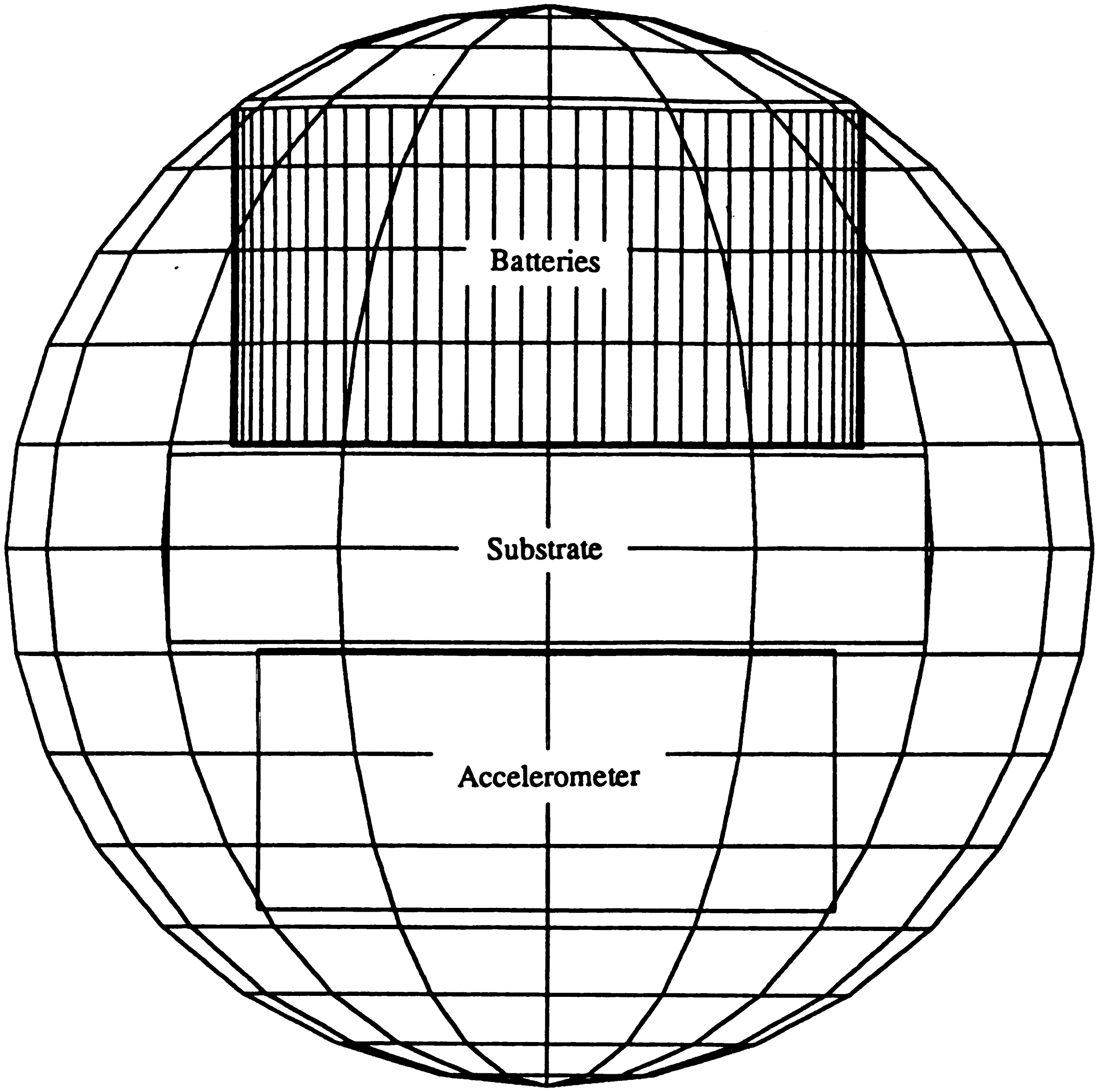


Figure 24 - The front view of the typical configuration for the sphere of 28.72mm (1.13 in.) diameter. Not included: rigid foam, beeswax, and epoxy.

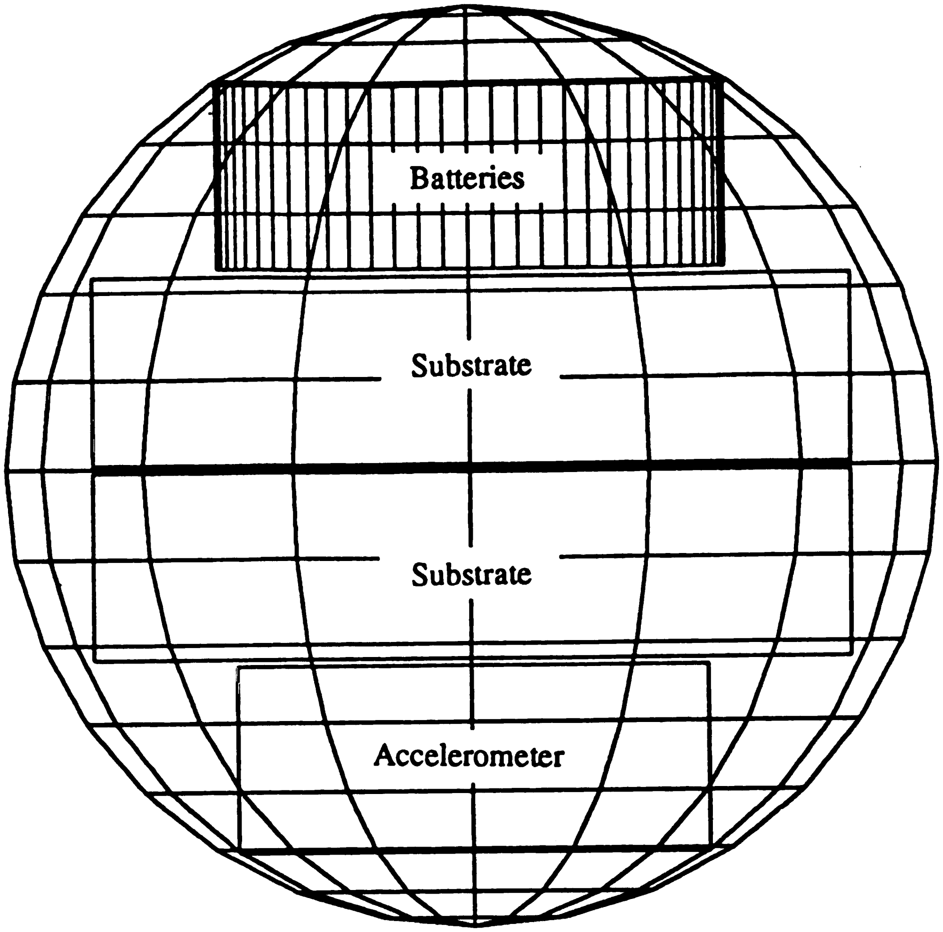


Figure 25 - The front view of the typical configuration for the sphere of 24.58 mm (0.97 in.) diameter. Not included: rigid foam, beeswax, and epoxy.

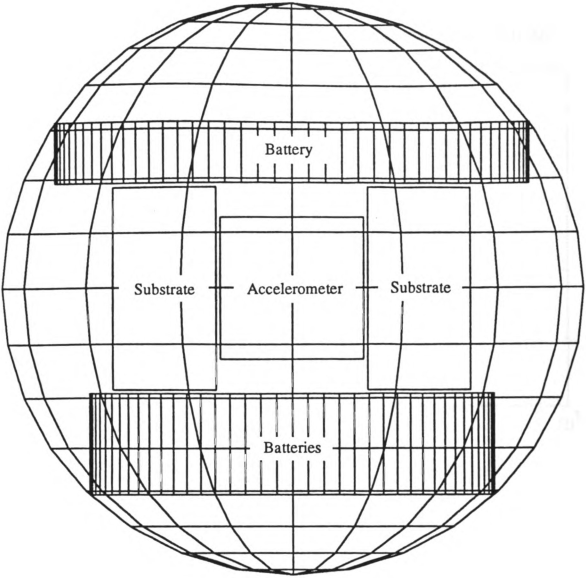


Figure 26 - The front view of the typical configuration for the sphere of 28.28 mm (1.12 in.) diameter. Not included: rigid foam, beeswax, and epoxy.

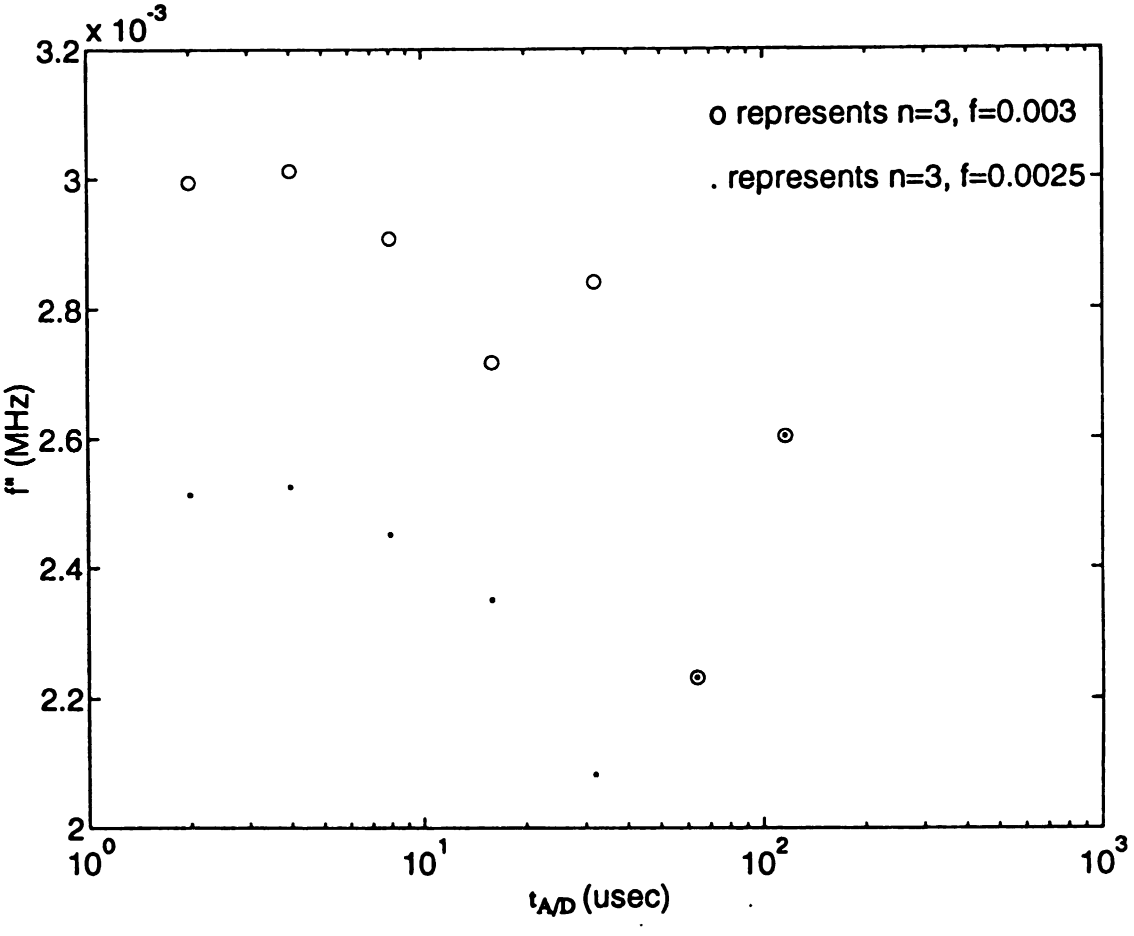


Figure 27 - f″ vs different specifications as given in Table 7.a.

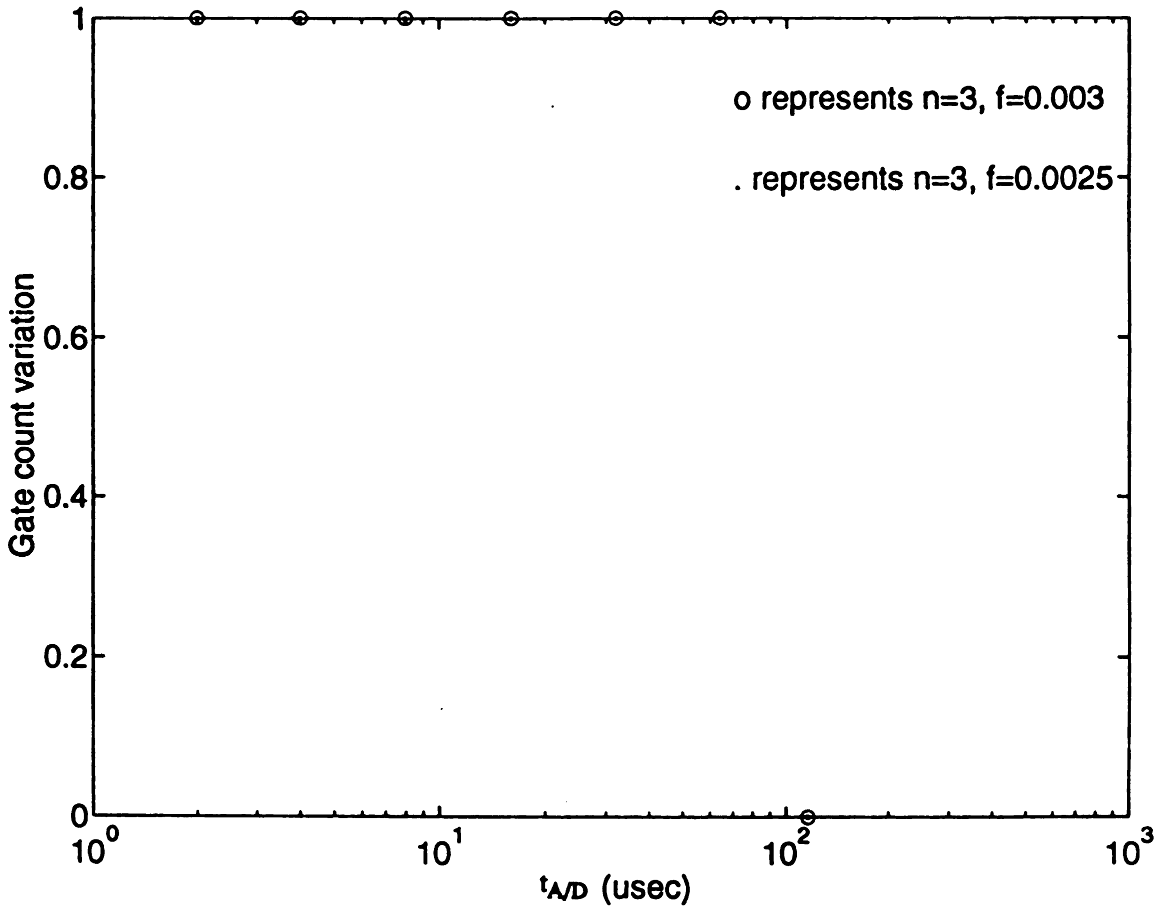


Figure 28 - Gate count variation vs different specifications as given in Table 7.b.

CHAPTER 5

Simulation Results

The development of a smaller IS design takes into consideration the experiences gained from the previous designs, specifically with respect the parameters of acceleration threshold, signal bandwidth and digital sampling rates. It has become clear that the units must be designed for specific applications in order to avoid dominance of data outside the realm of interest. For example, in transportation studies, the damaging vibrations and impacts occur at lower frequencies (less than 100 Hz, typically 5 to 25 Hz) than for packing line studies (more than 100 Hz). Thus, units designed to accept data to 4 kHz are saturated with useless data. For transportation studies, a 100 Hz or less low pass filter is mandatory, and a lower sampling rate consistent with this filter is required. Also, since the impacts experienced in transportation have lower peak acceleration levels, the sensitivity of the accelerometer must be increased and can be achieved by eliminating the shunt capacitor across each accelerometer. Based on the above conflicting constraints, no single electronics design will fulfill all application objectives. Thus, multiple electronics designs will be implemented to tailor the IS to the particular desired application. Each smart logic design will maintain fixed data acceptance thresholds and sampling rates. This is a major change from the μ-P based system where thresholds and sampling rates could be changed in software. The performance of new application designs can be studied using simulation techniques developed to investigate the properties and operation of the smart logic chip. Thus the simulation provides a powerful tool by which to evaluate the performance of new IS designs.

5.1 Architecture for IS Smart Logic Chip

It is desirable to view the operation of the entire smart logic device based on a realistic simulation. The overall system architecture for the smart logic composite chip is display in Figure19. The simulation is chosen to examine all facets of the operating system. Thus the data generated is extensive, so that only a small fraction can be shown and discussed. The detailed block diagram for the n-D data acquisition system with m restrictions is shown in Figure 2. The detailed circuit diagram for the data acquisition system with n=3, m=1 is shown in Figure 29.a. The detailed circuit diagram for the special case of 3-D and one restriction smart logic design is shown in Figure 29.b. It is this configuration for which extensive simulations have been carried out. The detailed architectural subsets for the components included in the simulation are shown in Figures 5, 8, 13, 16, and 18.

5.2 Simulation Results

We have used the VTI tools [64]~[69] for the circuit diagram layouts presented in this thesis. Some standard cells are normally stored in the VTI library, and we have used these to build larger and more complicated circuits, which are then connected together as shown in Figures 3, 5, 8, 13, 16, 18, 29.a and 29.b. After all the circuits were connected a simulation was developed, initially for the individual circuits and then for the global circuit in order to verify that both the system and subsystem performance is as expected.

For the simulation of the CSC generator circuit for n=3, m=1, t'=125 nsec, t=2 μsec, and f=0.003 MHz, each of the monitoring nodes are referenced in Figure 3. The SWC and CNTC are the output signals. The former is sent to the CTB as an input signal to generate S0, S1 and the latter is sent to COUNTER16 as an input signal to generate the address bus A15 to A0. The CLK8MH (t') and LE_ are input signals representing the clock from the crystal and latch enable signal, respectively. At first LE_=0, CLR (clear) is set to low in order to clear counter1 and RS is set to low in order to force RS1=0 which clears $Q_{15}Q_{14}...Q_1Q_0$ of counter2. Then CLR is set high and RS is set high which forces RS1=1, so that counter1 and $Q_{15}Q_{14}...Q_1Q_0$ of counter2 count pulses from CLK8MH (t′) and CLK500KH (t), respectively. T1 and T2 are signals obtained from the output of the AND gate and NAND gate, respectively and are important because they control the timing of the SWC. At first T1 is low and T2 is high, but after 328 μsec (2 μsec x $(2^7+2^5+2^2)$), T1 will go high to enable the SWC. After another 6 μsec (or 334 μsec from the origin given by 2 μsec x $(2^7+2^5+2^2+2^1+2^0)$), T2 will go low and force RS1=0 which will reset all outputs $Q_{15}Q_{14}...Q_1Q_0$ of counter2, and force T1 low resulting in a total high time of 6 μsec. Using T1 and CLK500KH, we can generate SWC to be used in the CTB. From SWC and LE_, we can get CNTC for counter addressing. The simulation results are shown in Figure 30.

For the simulation of the data acquisition system with n=3, m=1, t′=125 nsec, t=2 μsec, and f=0.003 MHz, each of the monitoring nodes are referenced in Figure 29.a. The O7,O6,....,O0 and A15,A14,...,A0 are the response signals. The former is sent to memory as input data and the latter is sent to memory as an input signal (address). The I7,I6,...,I0 and CLK8MH are input signals representing digitalized impact force and clock signal from the crystal, respectively. The important signals are S0, S1, LE_, RS1, SWC and CNTC because S0 and S1 choose the input signals from the x, y, z axis, LE_ controls CNTC, RS1 controls the SWC cycle, SWC generates S0 and S1, and CNTC generates address bus A15 to A0. The output signals are active only when SWC is high. S1 and S0 choose the input signals from the x, y, z axis according to the following assignments: S1,S0=00 chooses the x axis, S1,S0=01 chooses the y axis, and S1,S0=10 chooses the z axis. They continuously cycle to select the desired axial signal. Starting with I7=I6=1, I5=I4= ...=I0=0 or C0H which is an input signal above the threshold of 8AH, the output, delayed one time unit, gives O7=O6=1, O5=O4=....=O0=0 or C0H and the address lines A15,A14,...,A0 give a count of 3 to store the 8-bit x, y and z axis data into sequential memory. If I6 changes to 0 while the other data values stay fixed, the input signal takes a value of 80H which is below the threshold of 8AH so the output, delayed by one time unit (384 μsec), keeps the previous value of C0H and A15,A14, ...,A0 also keep their previous values (no data is written into memory). The simulation results for this operation are shown in Figure 31.

In Figure 9, the simulation results for the CSC generator circuit, for the special case n=3, m=1, t'=125 nsec, $t_{A/D}$ =116 μsec, and f=0.003 MHz, are shown. The detailed description of the operation is contained in Section 3.5.1. Figure 11 provides the simulation results of the CTB generator circuit for which the operation is outlined in Section 3.5.2. Figure 12 shows the simulation results for the latch circuit which is described in Section 3.5.3. Figure 15 provides the simulation results for the 8 bit comparator with predetermined thresholds. The detailed description of the comparator simulation is contained in Section 3.5.4. Finally, Figure 17 shows the simulation results for the 16-bit counter which is divided into four parts due to the amount of data displayed. The detailed description of the counter simulation is contained in Section 3.5.5.

For the typical simulation of the smart logic integrated circuit, for the special case n=3, m=1, t'=125 nsec, $t_{A/D}$=116 μsec, and f=0.003 MHz, each of the monitoring nodes are identified in Figure 29.b. The O7 to O0 and A15 to A0 are output signals. The former is sent to memory as an input data signal and the latter is sent to memory as an input address signal. The I7 to I0 and CLK8MH are input signals representing digitalized impact force data and clock signals from the crystal, respectively. The important signals are S0, S1, LE_, RS1, SWC and CNTC because S0 and S1 choose the input signals from the x, y, z axis, LE_ controls the CNTC, RS1 controls the SWC cycle, SWC generates S0 and S1, and CNTC generates the address bus A15 to A0. The detailed descriptions of these signals are contained in Section 5.3. The simulation results are shown in Figure 32.

5.3 Discussion

The relative input and output data are listed in Appendix 4. The corresponding signal plots are shown in Figure 32. The simulation trace data file of the SLIC are listed in the Appendix 5. The input signals are two impact forces shown in Figure 33. The output signals are delayed one time unit (384 μs) and the first two data values are replaced by a heading and time of occurrence for each significant impact pulse. Because the sampling time is very small compared to each impact period, generally, each impact pulse has many samples, so that the first two data points can easily be sacrified. Because simulation times are very long for each data sample, we only choose 4 data samples for the first impact force and 6 data samples for the second impact.

The first input data for the first impact is 84H, 84H, 84H which is below the threshold. The relative output data which is delayed one time unit is arbitrary. The second input data is 84H, 84H, C0H which is over the threshold. The relative output data which is delayed one time unit is 80H, 80H, 80H which represents heading. The third input data is E0H, E0H, E0H which is also over the threshold. The relative output data which is delayed one time unit is 00H, 00H, 0CH which represents the real time clock count. The fourth input data is E0H, E0H, 84H which is over the threshold. The relative output data which is delayed one time unit is E0H, E0H, 84H, the same as the input data.

The first input data for the second input pulse is 84H, 84H, 84H which is below the threshold. The relative output data which is delayed one time unit is arbitrary. The second input data is 84H, C0H, C0H which is over the threshold. The relative output data, delayed one time unit, is 80H, 80H, 80H which represents heading. The third input data is A0H, A0H, A0H which is over the threshold. The relative output data, delayed one time unit, is 00H, 00H, 18H which represents the real time clock count. The fourth input data is E0H, E0H, E0H, which is over the threshold, gives an output, delayed one time unit, of E0H, E0H, E0H which is the same as the input data. The fifth input data above threshold, C0H, C0H, C0H gives a unit delayed output of C0H, C0H, C0H again the same as the input data. The sixth input data is 84H, 84H, 84H which is below the threshold, and thus gives a unit delayed output which is arbitrary.

Thus the system processes all the input signals which are above the threshold and displays them as output signals, except for the first two data points above threshold which are used to generate a heading and time before each group of output signals relative to one impact. Thus we can identify each group of output signals for each impact pulse and save memory by ignoring all input signals below the threshold. We can recognize the time of occurrance for each impact.

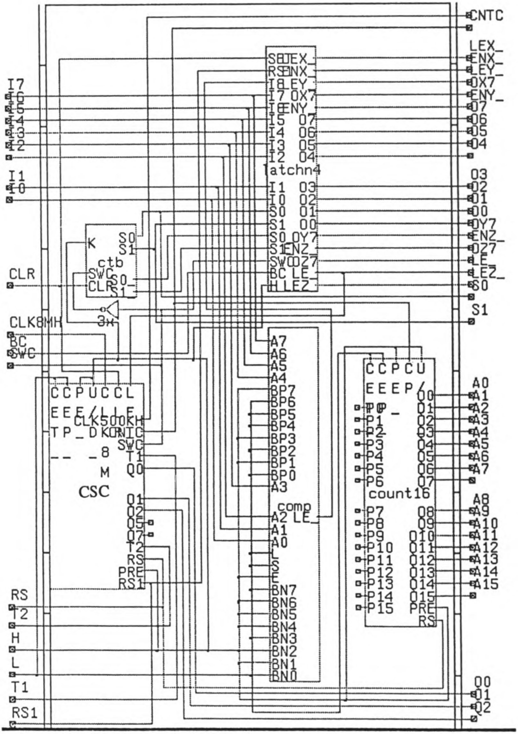


Figure 29.a - The detailed circuit diagram for the data acquisition system with n=3, m=1.

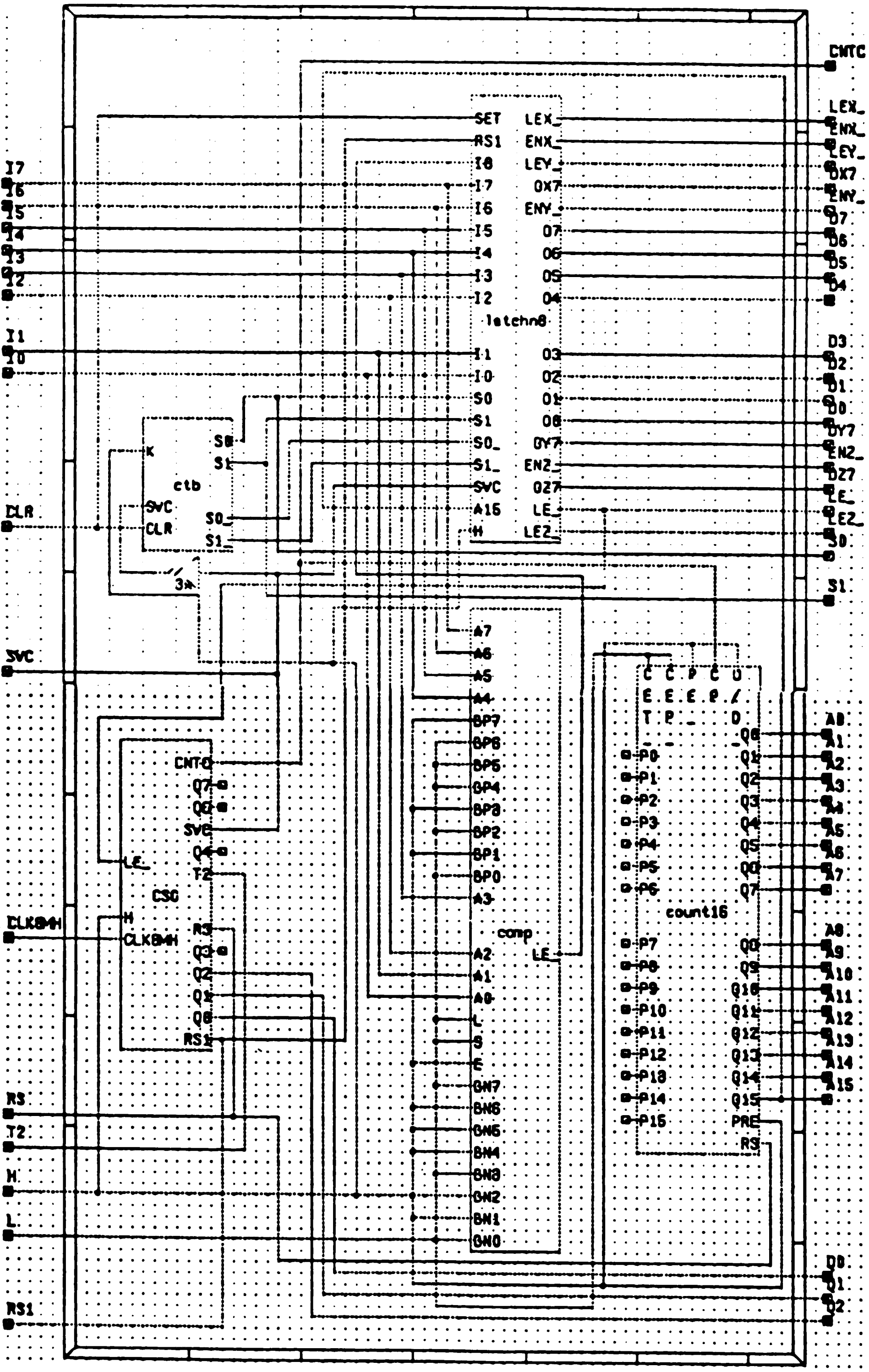


Figure 29.b - The detailed circuit diagram for the smart logic circuit (special case of n=3, m=1).

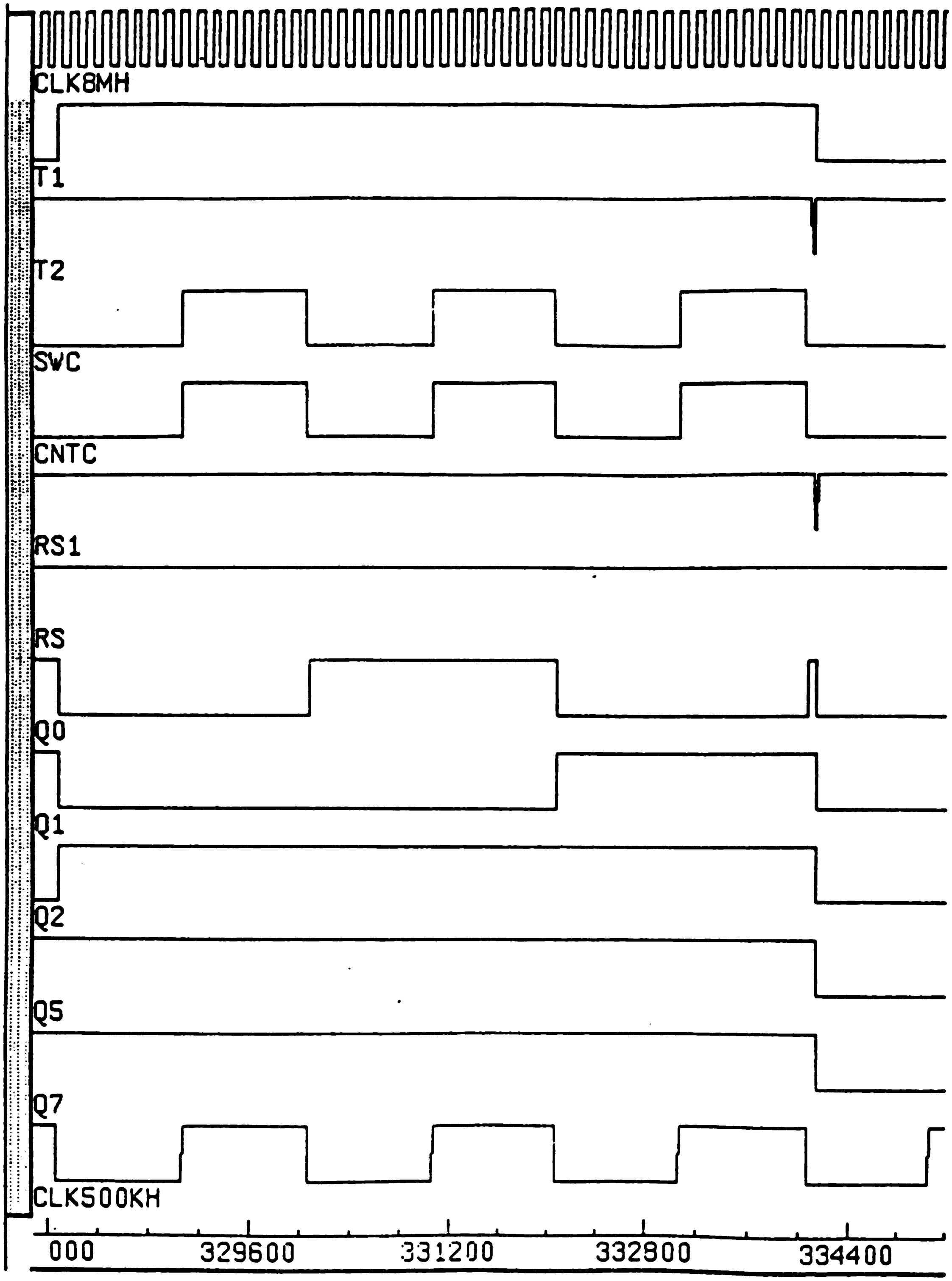


Figure 30 - The simulation results of the CSC (counter & switch clock) generator circuit for n=3.

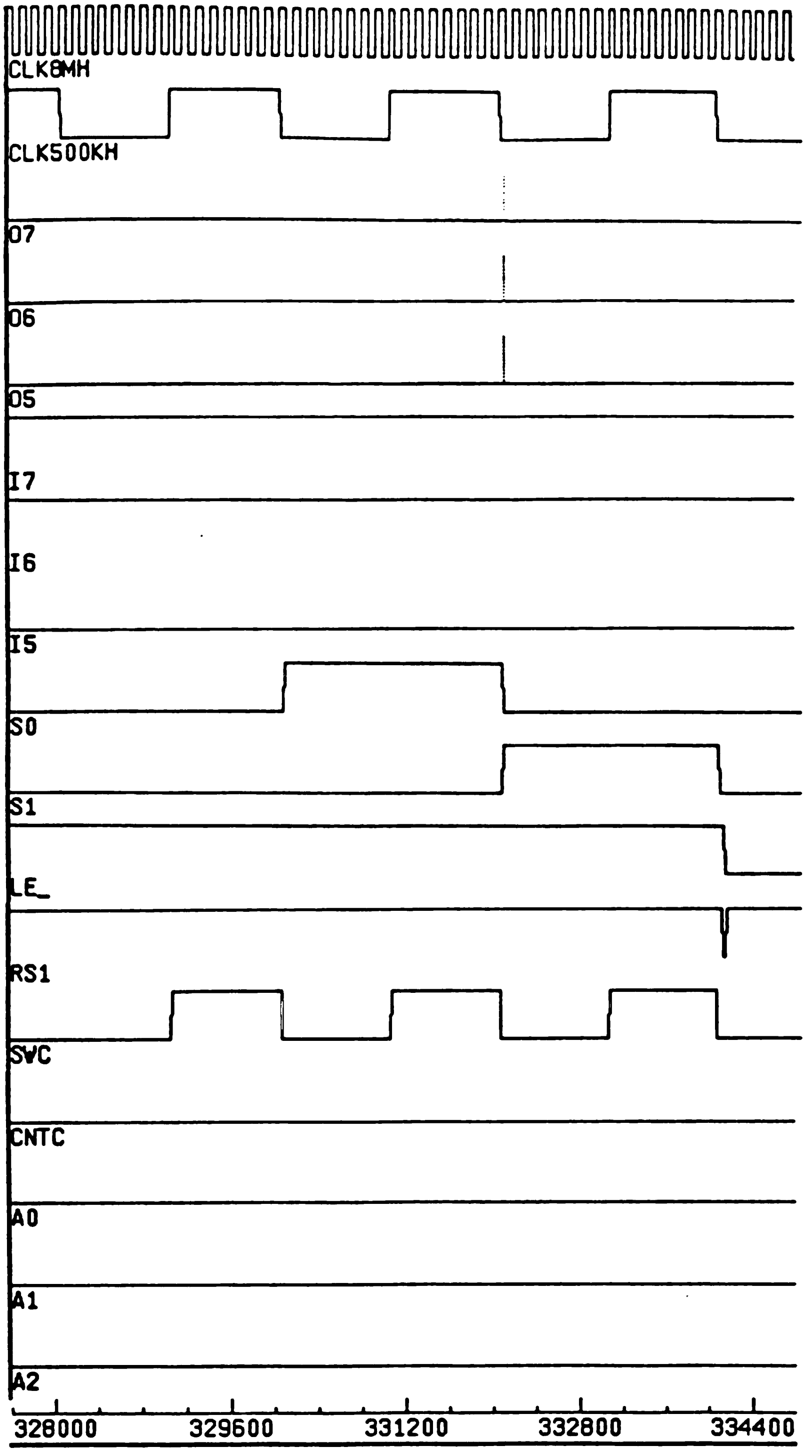


Figure 31.a - The simulation results of the data acquisition system with n=3, m=1, t′=125 nsec, t=2 μsec, f=0.003 MHz.

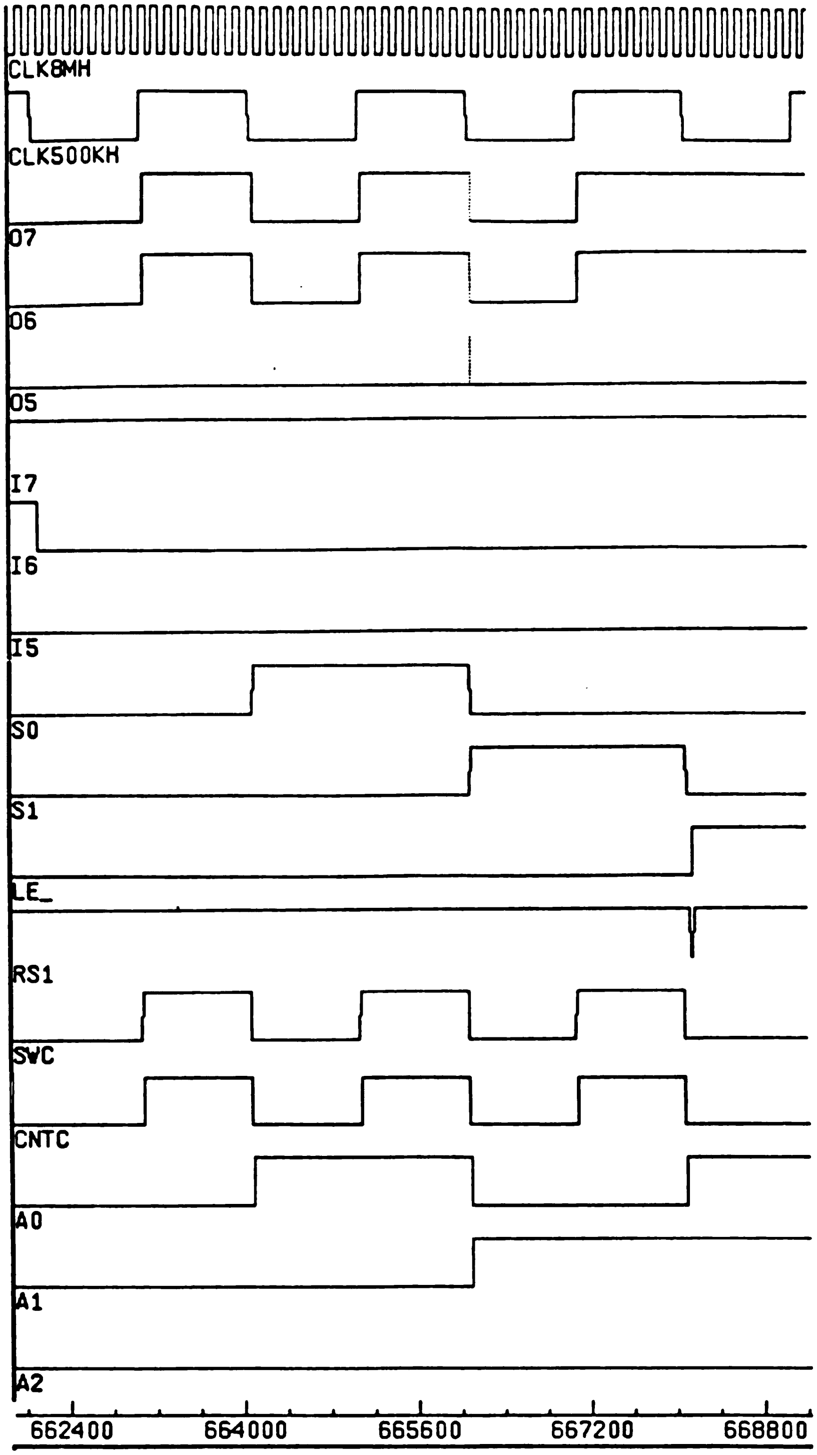


Figure 31.b - The simulation results of the data acquisition system with n=3, m=1, t′=125 nsec, t=2 μsec, f=0.003 MHz, continued.

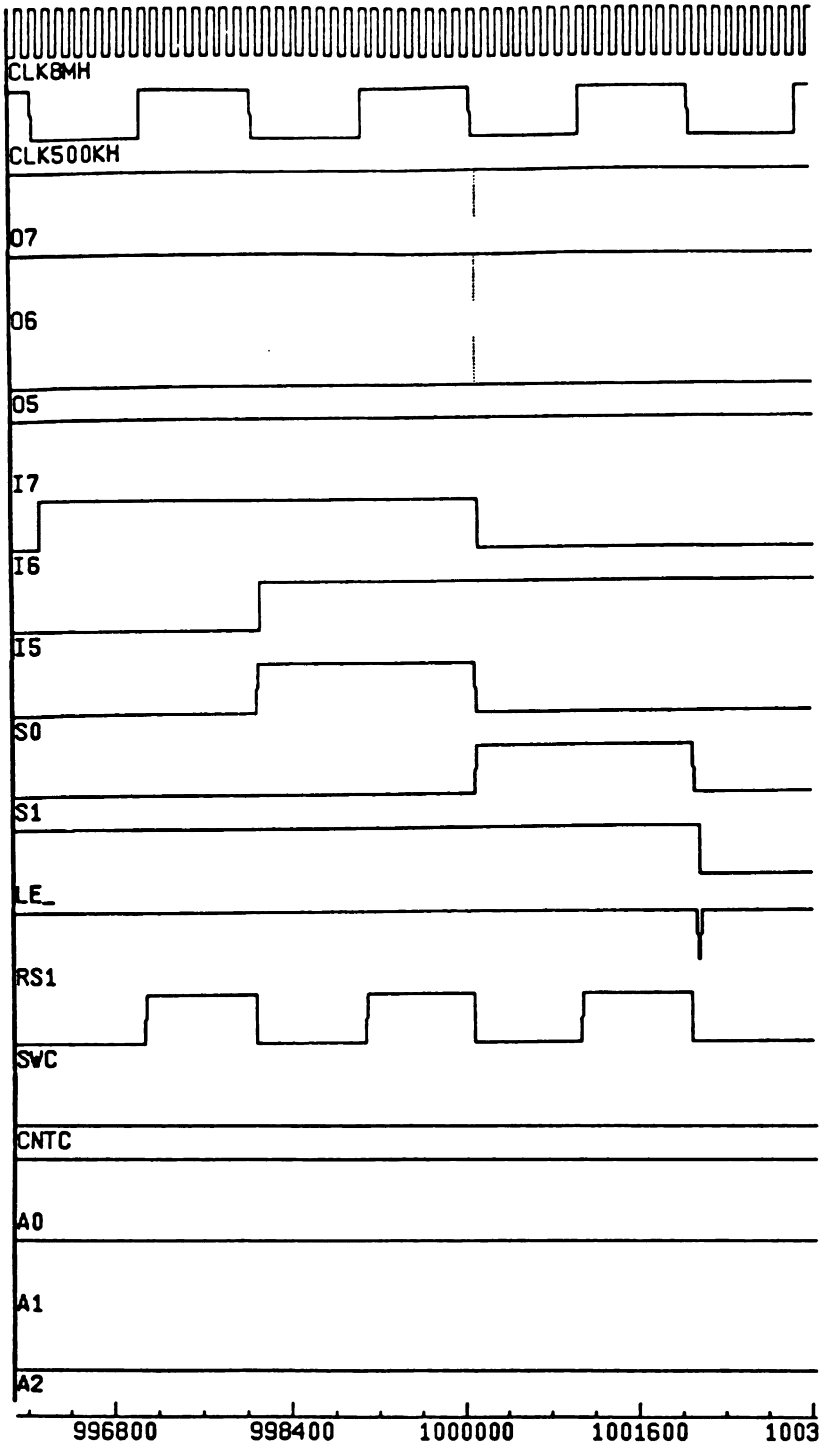


Figure 31.c - The simulation results of the data acquisition system with n=3, m=1, t′=125 nsec, t=2 μsec, f=0.003 MHz, continued.

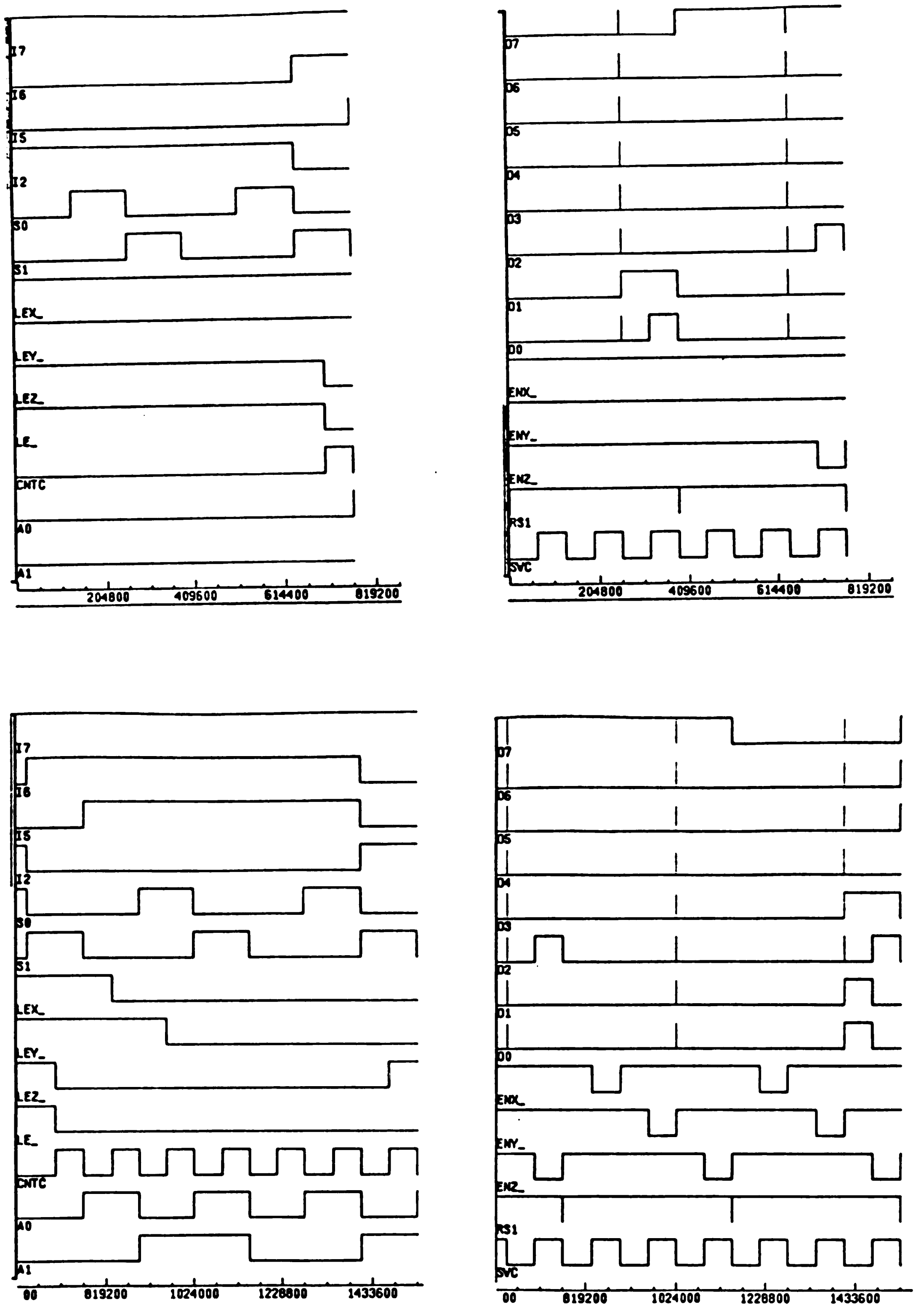


Figure 32.a - The typical simulation results of the smart logic integrated circuit (special case of n=3, m=1).

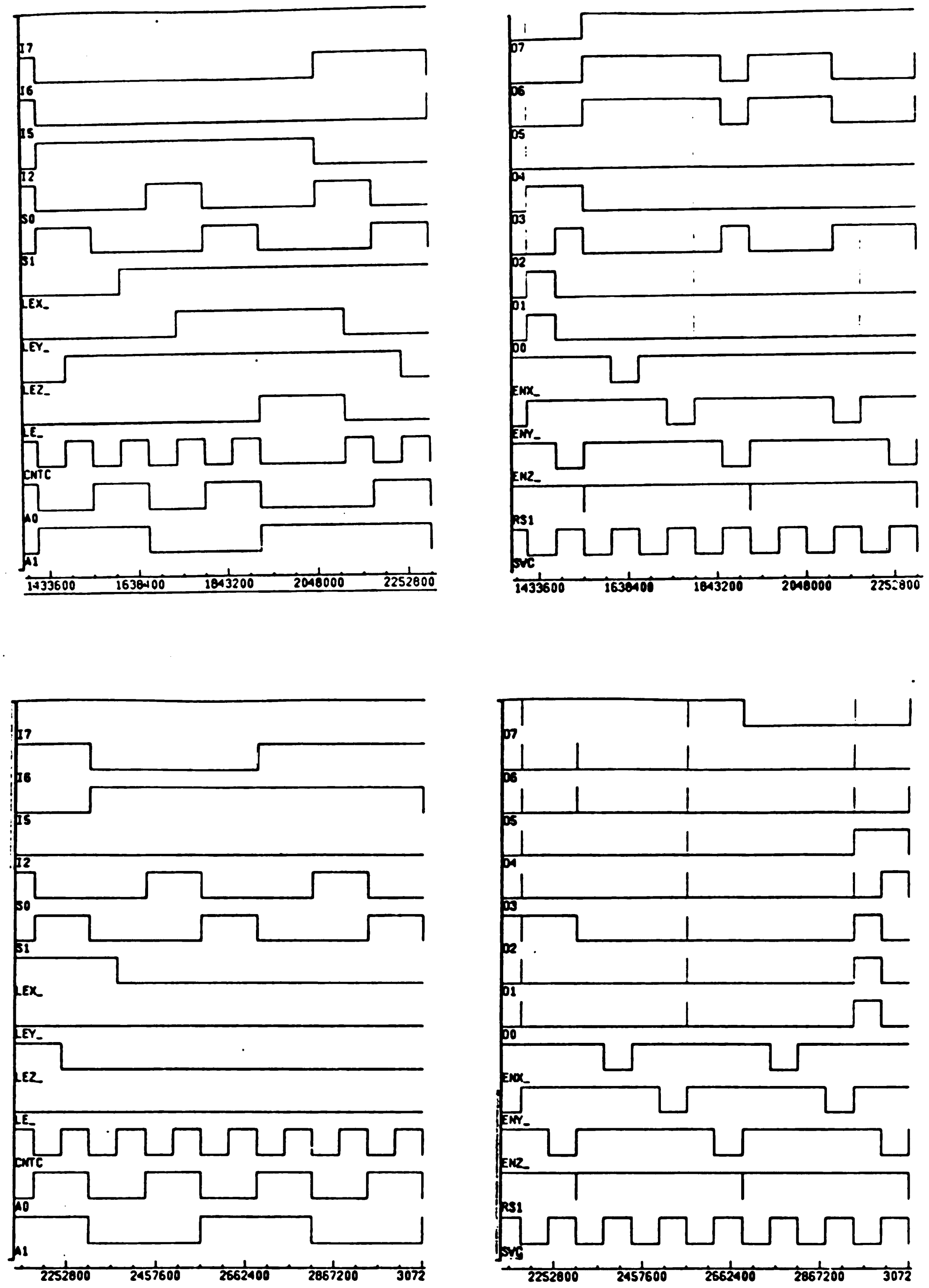


Figure 32.b - The typical simulation results of the smart logic integrated circuit (special case of n=3, m=1), continued.

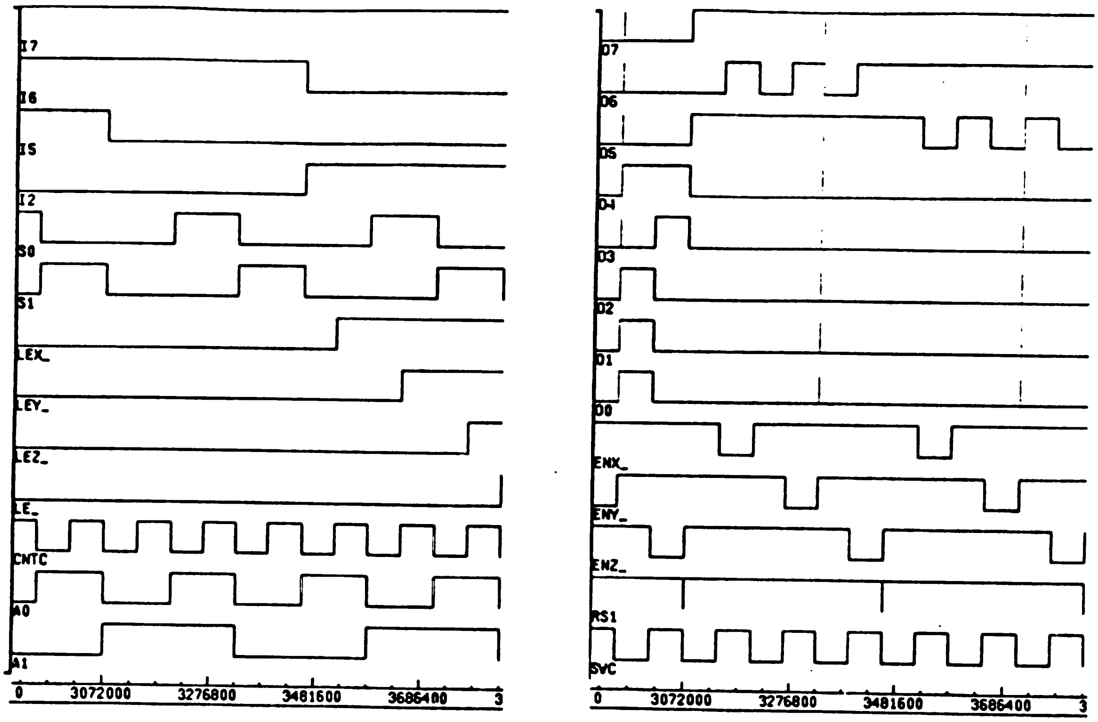


Figure 32.c - The typical simulation results of the smart logic integrated circuit (special case of n=3, m=1), continued.

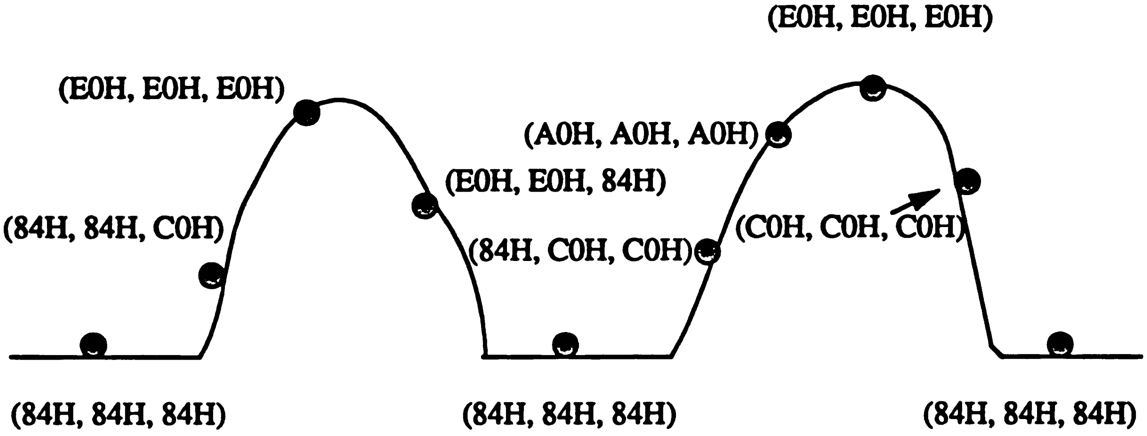


Figure 33 - Two impact forces which contain ten sampled data.

CHAPTER 6

Conclusions and Directions for Future Work

6.1 Conclusions

The major motivation for the development of smart logic is to replace the existing μ-P circuit in order to achieve a total data acquisition sphere of dimension less than 50.8 mm (2 in.). Other objectives in the research are to reduce power and cost. An extensive industrial demand for such a device exists.

As discussed in Chapter 4, the size of the composite smart logic chip based IS is about 38 mm (1.5 in.), or 1/12 the volume of the first commercial unit. The power consumption for the composite smart logic chip based IS is about 5.24 mA per hour at Vcc=5 V, which is less than half the current consumption of the conventional μ-P based circuit. Thus the smart logic configuration can accomodate all of the desired features - smaller size, longer operation time and ultimately lower cost.

If specifications were to change, for example $t_{A/D}$ changed from 2 μsec to 116 μsec, with the other parameters fixed, the power consumption and size remain approximately the same. This is true even if the other parameters are varied, as shown and discussed in Chapter 4. Thus the variation in power and size vs specification change is very small indicating a very robost system.

Based on the simulation results discussed in Chapter 5, the smart logic system can accept all the input signals above an assigned threshold, store these in memory, along with a data header, and output the data when requested. Thus every impact is recorded and identified if it surpasses its assigned threshold but ignored if it falls below the threshold. A 3-D smart logic data acquisition system design with one restriction is developed.

6.2 Directions for Future Work

A composite smart logic chip based IS is a good choice for reducing the size of the IS and thus allowing the measurement of impacts to smaller produce. This process can be expanded to include low voltage ICs for example, 3 V as proposed by Prince and Salters [62] and Williams [63], or to pursue the Fujitsu proposal to operate a 32 K byte full CMOS static RAM down to 1 V. These approaches will allow the battery requirements to decrease, the three cells may be reduced to two cells or one cell, and the battery will last longer between recharges. Other benefits include substantially smaller size and lower heat dissipation than for 5 V systems. An early approach would be to fabricate a SLIC as a die and bond four dice - Op Amp, A/D, RAM, SLIC onto a substrate as shown in Figure 20. A diamond film substrate would provide an ideal heat sink for such a high density circuit configuration. The accelerometer and battery could be connected in a configuration such as shown in Figure 23.

APPENDICES

Appendix 1 Simulation Results for the Latch

The input signals used to run the simulation consist of 2 hexadecimal characters which represent an impact force similar to that shown in Figure 14. The output signals are delayed one time unit (384 μs) and the first two data values are replaced by a heading and time of occurrence for any significant impact pulse (impact data samples above the threshold). High/Low threshold is set at 8AH/76H (equivalent to +10 g/-10 g). "O" represents acknowledgment of a signal above the threshold and "X" represents a signal below the threshold.

Input signal: placed on the I7~I0 binary data lines

84H, 84H, 84H (10000100B, 10000100B, 10000100B) X

84H, 84H, C0H (10000100B, 10000100B, 11000000B) O

E0H, E0H, E0H (11100000B, 11100000B, 11100000B) O

E0H, E0H, 84H (11100000B, 11100000B, 10000100B) O

84H, 84H, 84H (10000100B, 10000100B, 10000100B) X

Output signal: received on the O7~O0 binary data lines

80H, 80H, 84H (10000000B, 10000000B, 10000100B) Arbitrary

80H, 80H, 80H (10000000B, 10000000B, 10000000B) Heading

00H, 00H, 0CH (00000000B, 00000000B, 00001100B) Time

E0H, E0H, 84H (11100000B, 11100000B, 10000100B) O

E0H, 84H, 84H (11100000B, 10000100B, 10000100B) Arbitrary

Appendix 2 The Die Size Variation of the SLIC vs Specification Change

The $t_{A/D}$ represents the maximum A/D conversion time, t' represents the internal clock period generated from the crystal, f represents the sampling frequency, n represents the data dimension, t represents the duration of the clock pulses in the burst, 1/tf represents the round off of clock count per sampling frequency, $1/tf''$ represents the clock count per actual sampling frequency and f'' represents the actual sampling frequency. The specification changes are made to $t_{A/D}$ and f. No change in die size occurs as a result of specification changes.

$t_{A/D}$	t'	f	n	1/tf		$1/tf''$	f''	die size variation
µsec	µsec	MHz		decimal	binary	binary	MHz	mm^2
2	0.125	0.003	3	167	10100111	10100111	0.002994	0
4	0.125	0.003	3	83	01010011	01010011	0.003012	0
8	0.125	0.003	3	42	00101010	00101011	0.002907	0
16	0.125	0.003	3	21	00010101	00010111	0.002717	0
32	0.125	0.003	3	10	00001010	00001011	0.002841	0
64	0.125	0.003	3	5	00000101	00000111	0.002232	0
116	0.125	0.003	3	3	00000011	00000011	0.002604	0
2	0.125	0.0025	3	200	11001000	11000111	0.002513	0
4	0.125	0.0025	3	100	01100100	01100011	0.002525	0
8	0.125	0.0025	3	50	00110010	00110011	0.002451	0
16	0.125	0.0025	3	25	00011001	00011011	0.002315	0
32	0.125	0.0025	3	13	00001101	00001111	0.002083	0
64	0.125	0.0025	3	6	00000110	00000111	0.002232	0
116	0.125	0.0025	3	3	00000011	00000011	0.002604	0

Appendix 3 The Chip Size Variation of the Composite Smart Logic Chip vs Specification Change

The $t_{A/D}$ represents the maximum A/D conversion time, t' represents the internal clock period generated from the crystal, f represents the sampling frequency, n represents the data dimension, t represents the duration of the clock pulses in the burst, 1/tf represents the round off of clock count per sampling frequency, $1/tf''$ represents the clock count per actual sampling frequency and f'' represents the actual sampling frequency. The specification changes are made to $t_{A/D}$ and f. No change in chip size occurs as a result of specification changes.

$t_{A/D}$	t'	f	n	1/tf		$1/tf''$	f''	chip size variation
μsec	μsec	MHz		decimal	binary	binary	MHz	mm^2
2	0.125	0.003	3	167	10100111	10100111	0.002994	0
4	0.125	0.003	3	83	01010011	01010011	0.003012	0
8	0.125	0.003	3	42	00101010	00101011	0.002907	0
16	0.125	0.003	3	21	00010101	00010111	0.002717	0
32	0.125	0.003	3	10	00001010	00001011	0.002841	0
64	0.125	0.003	3	5	00000101	00000111	0.002232	0
116	0.125	0.003	3	3	00000011	00000011	0.002604	0
2	0.125	0.0025	3	200	11001000	11000111	0.002513	0
4	0.125	0.0025	3	100	01100100	01100011	0.002525	0
8	0.125	0.0025	3	50	00110010	00110011	0.002451	0
16	0.125	0.0025	3	25	00011001	00011011	0.002315	0
32	0.125	0.0025	3	13	00001101	00001111	0.002083	0
64	0.125	0.0025	3	6	00000110	00000111	0.002232	0
116	0.125	0.0025	3	3	00000011	00000011	0.002604	0

Appendix 4 Simulation Results of the SLIC

The input signals used to run the simulation consist of 2 hexadecimal characters which represent two impact forces similar to that shown in Figure 33. The output signals are delayed one time unit (384 μs) and the first two data values are replaced by a heading and time of occurrence for any significant impact pulse (impact data samples above the threshold). High/Low threshold is set at 8AH/76H (equivalent to +10 g/-10 g). "O" represents acknowledgment of a signal above the threshold and "X" represents a signal below the threshold.

Input signal: placed on the I7~I0 binary data lines

84H, 84H, 84H (10000100B, 10000100B, 10000100B) X

84H, 84H, C0H (10000100B, 10000100B, 11000000B) O

E0H, E0H, E0H (11100000B, 11100000B, 11100000B) O

E0H, E0H, 84H (11100000B, 11100000B, 10000100B) O

84H, 84H, 84H (10000100B, 10000100B, 10000100B) X

84H, C0H, C0H (10000100B, 11000000B, 11000000B) O

A0H, A0H, A0H (10100000B, 10100000B, 10100000B) O

E0H, E0H, E0H (11100000B, 11100000B, 11100000B) O

C0H, C0H, C0H (11000000B, 11000000B, 11000000B) O

84H, 84H, 84H (10000100B, 10000100B, 10000100B) X

Output signal: received on the O7~O0 binary data lines

80H, 80H, 84H (10000000B, 10000000B, 10000100B) Arbitrary

80H, 80H, 80H (10000000B, 10000000B, 10000000B) Heading

00H, 00H, 0CH (00000000B, 00000000B, 00001100B) Time

E0H, E0H, 84H (11100000B, 11100000B, 10000100B) O

E0H, 84H, 84H (11100000B, 10000100B, 10000100B) Arbitrary

80H, 80H, 80H (10000000B, 10000000B, 10000000B) Heading

00H, 00H, 18H (00000000B, 00000000B, 00011000B) Time

E0H, E0H, E0H (11100000B, 11100000B, 11100000B) O

C0H, C0H, C0H (11000000B, 11000000B, 11000000B) O

Address for memory: A14~A0 binary data address lines

The simulation monitoring nodes shown in Figure 32 are:

1. O7, O6, O5, O4, O3, O2, O1, O0: Output data bit 7, 6, 5, 4, 3, 2, 1, 0 from latch.
2. ENX_, ENY_, ENZ_: Post latch enable control for X, Y, Z axis.
3. RS1: Reset signal in CSC (counter & switch clock) generator.
4. SWC: Switch clock generated from CSC.
5. I7, I6, I5, I2: Input data bit 7, 6, 5, 2 to pre-latch.
6. S0, S1: Selective address for A/D analog switch.
7. LEX_, LEY_, LEZ_: Latch enable control from comparator for X, Y, Z axis.
8. LE_: Post latch enable control.
9. CNTC: Counter clock generated from CSC.
10. A0, A1: Address bus for memory.

Appendix 5 Simulation Trace Data File of the SLIC

The input signals used to run the simulation consist of 2 hexadecimal characters which represent two impact forces similar to that shown in Figure 33. The output signals are delayed one time unit (384 μs) and the first two data values are replaced by a heading and time of occurrence for any significant impact pulse (impact data samples above the threshold). High/Low threshold is set at 8AH/76H (equivalent to +10 g/-10 g).

```
Time = 1:1    [0 ns]
                   I7  =  1    [0]  (input)(display)
                   I6  =  0    [0]  (input)(display)
                   I5  =  0    [0]  (input)(display)
                   I2  =  1    [0]  (input)(display)
                   S0  =  u    [0]  (output)(display)
                   S1  =  u    [0]  (output)(display)
                 LEX_  =  u    [0]  (output)(display)
                 LEY_  =  u    [0]  (output)(display)
                 LEZ_  =  u    [0]  (output)(display)
                  LE_  =  u    [0]  (output)(display)
                 CNTC  =  u    [0]  (output)(display)
                   A0  =  u    [0]  (output)(display)
                   A1  =  u    [0]  (output)(display)
                   A2  =  u    [0]  (output)(display)
                   A3  =  u    [0]  (output)(display)
                   A4  =  u    [0]  (output)(display)
                   I4  =  0    [0]  (input)(display)
                   I3  =  0    [0]  (input)(display)
                   I1  =  0    [0]  (input)(display)
                   I0  =  0    [0]  (input)(display)
                   T2  =  u    [0]  (output)(display)
                   Q0  =  u    [0]  (output)(display)
                   Q1  =  u    [0]  (output)(display)
                   Q2  =  u    [0]  (output)(display)
                  CLR  =  0    [0]  (input)(display)
                   RS  =  0    [0]  (input)(display)
                  OX7  =  u    [0]  (output)(display)
                  OY7  =  u    [0]  (output)(display)
                  OZ7  =  u    [0]  (output)(display)
               CLK8MH  =  u    [0]  (clock)(display)
                   O7  =  u    [0]  (output)(display)
                   O6  =  u    [0]  (output)(display)
                   O5  =  u    [0]  (output)(display)
                   O4  =  u    [0]  (output)(display)
                   O3  =  u    [0]  (output)(display)
                   O2  =  u    [0]  (output)(display)
                   O1  =  u    [0]  (output)(display)
                   O0  =  u    [0]  (output)(display)
                 ENX_  =  u    [0]  (output)(display)
                 ENY_  =  u    [0]  (output)(display)
                 ENZ_  =  u    [0]  (output)(display)
                  RS1  =  u    [0]  (output)(display)
                  SWC  =  u    [0]  (output)(display)
```

```
  Time = 1:1   [0 ns]
CLK8MH --> 0     [+0/0]
LEZ_ --> 1     [+4.1/0]
LEY_ --> 1     [+4.1/0]
LEX_ --> 1     [+4.1/0]
RS1 --> 0     [+7/0]
ENY_ --> 1     [+13.3/0]
ENZ_ --> 1     [+13.3/0]
A2 --> 0     [+15/0]
A3 --> 0     [+15/0]
A0 --> 0     [+15/0]
A1 --> 0     [+15/0]
A4 --> 0     [+15/0]
T2 --> 1     [+17.6/0]
OX7 --> 1     [+17.7/0]
OY7 --> 1     [+17.7/0]
OZ7 --> 1     [+17.7/0]
Q2 --> 0     [+19.5/0]
LE_ --> 0     [+20.4/0]
Q1 --> 0     [+20.6/0]
Q0 --> 0     [+22.1/0]
S1 --> 0     [+24.4/0]
SWC --> 0     [+26.8/0]
CNTC --> 0     [+29.4/0]
S0 --> 0     [+33.7/0]
ENX_ --> 1     [+35.3/0]
O6 --> 0     [+36.6/0]
O3 --> 0     [+36.6/0]
O2 --> 0     [+36.6/0]
O1 --> 0     [+36.6/0]
O0 --> 0     [+36.6/0]
O4 --> 0     [+36.6/0]
O5 --> 0     [+36.6/0]
O7 --> 1     [+45.2/0]
  Time = 1:2   [62.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 2:1   [125 ns]
RS --> 1     [+0/0]
CLR --> 1     [+0/0]
CLK8MH --> 0     [+0/0]
RS1 --> 1     [+6.4/6.4]
LE_ --> 1     [+24.7/5.1]
O7 --> 0     [+32/.4]
  Time = 2:2   [187.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 3:1   [250 ns]
CLK8MH --> 0     [+0/0]
  Time = 3:2   [312.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 4:1   [375 ns]
CLK8MH --> 0     [+0/0]
  Time = 4:2   [437.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 5:1   [500 ns]
CLK8MH --> 0     [+0/0]
  Time = 5:2   [562.5 ns]
CLK8MH --> 1     [+0/0]
```

```
  Time = 5121:1    [640000 ns]
                            I7  =  1    [0]  (input)(display)
                            I6  =  0    [0]  (input)(display)
                            I5  =  0    [0]  (input)(display)
                            I2  =  1    [0]  (input)(display)
                            S0  =  1    [512131.1]  (output)(display)
                            S1  =  0    [384128.1]  (output)(display)
                          LEX_  =  1    [4.1]  (output)(display)
                          LEY_  =  1    [4.1]  (output)(display)
                          LEZ_  =  1    [4.1]  (output)(display)
                           LE_  =  1    [149.7]  (output)(display)
                          CNTC  =  0    [29.4]  (output)(display)
                            A0  =  0    [15]  (output)(display)
                            A1  =  0    [15]  (output)(display)
                            A2  =  0    [15]  (output)(display)
                            A3  =  0    [15]  (output)(display)
                            A4  =  0    [15]  (output)(display)
                            I4  =  0    [0]  (input)(display)
                            I3  =  0    [0]  (input)(display)
                            I1  =  0    [0]  (input)(display)
                            I0  =  0    [0]  (input)(display)
                            T2  =  1    [384145.1]  (output)(display)
                            Q0  =  1    [638092.6]  (output)(display)
                            Q1  =  1    [636091]  (output)(display)
                            Q2  =  1    [632089.7]  (output)(display)
                           CLR  =  1    [125]  (input)(display)
                            RS  =  1    [125]  (input)(display)
                           OX7  =  1    [17.7]  (output)(display)
                           OY7  =  1    [17.7]  (output)(display)
                           OZ7  =  1    [17.7]  (output)(display)
                        CLK8MH  =  1    [639937.5]  (clock)(display)
                            O7  =  1    [384176.6]  (output)(display)
                            O6  =  0    [256150.2]  (output)(display)
                            O5  =  0    [256150.2]  (output)(display)
                            O4  =  0    [256150.2]  (output)(display)
                            O3  =  0    [256150.2]  (output)(display)
                            O2  =  0    [256150.2]  (output)(display)
                            O1  =  0    [384140.3]  (output)(display)
                            O0  =  0    [384140.3]  (output)(display)
                          ENX_  =  1    [35.3]  (output)(display)
                          ENY_  =  1    [13.3]  (output)(display)
                          ENZ_  =  1    [13.3]  (output)(display)
                           RS1  =  1    [384151.5]  (output)(display)
                           SWC  =  1    [576098]  (output)(display)
  Time = 5121:1    [640000 ns]
CLK8MH --> 0     [+0/0]
  Time = 5121:2    [640062.5 ns]
CLK8MH --> 1     [+0/0]
Q2 --> 0     [+25.5/7.9]
Q1 --> 0     [+26.7/9.1]
Q0 --> 0     [+28.2/10.6]
SWC --> 0     [+33/15.4]
S1 --> 1     [+59.7/17.4]
```

```
  Time = 5122:1   [640125 ns]
CLK8MH --> 0     [+0/0]
O6 --> u     [+8.7/0]
O3 --> u     [+8.7/0]
O2 --> u     [+8.7/0]
O1 --> u     [+8.7/0]
O0 --> u     [+8.7/0]
O4 --> u     [+8.7/0]
O5 --> u     [+8.7/0]
O7 --> u     [+9.4/0]
S0 --> 0     [+13/29.2]
O7 --> 1     [+24.5/0]
O6 --> 0     [+25.2/0]
O3 --> 0     [+25.2/0]
O2 --> 0     [+25.2/0]
O1 --> 0     [+25.2/0]
O0 --> 0     [+25.2/0]
O4 --> 0     [+25.2/0]
O5 --> 0     [+25.2/0]
  Time = 5122:2   [640187.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 5123:1   [640250 ns]
CLK8MH --> 0     [+0/0]
  Time = 5123:2   [640312.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 5124:1   [640375 ns]
CLK8MH --> 0     [+0/0]
  Time = 5124:2   [640437.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 5125:1   [640500 ns]
I6 --> 1     [+0/0]
I2 --> 0     [+0/0]
CLK8MH --> 0     [+0/0]
  Time = 5125:2   [640562.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 5126:1   [640625 ns]
CLK8MH --> 0     [+0/0]
  Time = 5126:2   [640687.5 ns]
CLK8MH --> 1     [+0/0]
```

```
  Time = 6138:1   [767125 ns]
                                     I7  =  1    [0]  (input)(display)
                                     I6  =  1    [640500]  (input)(display)
                                     I5  =  0    [0]  (input)(display)
                                     I2  =  0    [640500]  (input)(display)
                                     S0  =  0    [640138]  (output)(display)
                                     S1  =  1    [640122.2]  (output)(display)
                                   LEX_  =  1    [4.1]  (output)(display)
                                   LEY_  =  1    [4.1]  (output)(display)
                                   LEZ_  =  0    [704113.3]  (output)(display)
                                    LE_  =  0    [704125.6]  (output)(display)
                                   CNTC  =  1    [704128.2]  (output)(display)
                                     A0  =  0    [15]  (output)(display)
                                     A1  =  0    [15]  (output)(display)
                                     A2  =  0    [15]  (output)(display)
                                     A3  =  0    [15]  (output)(display)
                                     A4  =  0    [15]  (output)(display)
                                     I4  =  0    [0]  (input)(display)
                                     I3  =  0    [0]  (input)(display)
                                     I1  =  0    [0]  (input)(display)
                                     I0  =  0    [0]  (input)(display)
                                     T2  =  1    [384145.1]  (output)(display)
                                     Q0  =  1    [766092.6]  (output)(display)
                                     Q1  =  1    [764091]  (output)(display)
                                     Q2  =  1    [760089.7]  (output)(display)
                                    CLR  =  1    [125]  (input)(display)
                                     RS  =  1    [125]  (input)(display)
                                    OX7  =  1    [17.7]  (output)(display)
                                    OY7  =  1    [17.7]  (output)(display)
                                    OZ7  =  1    [17.7]  (output)(display)
                                 CLK8MH  =  1    [767062.5]  (clock)(display)
                                     O7  =  1    [640149.5]  (output)(display)
                                     O6  =  0    [640150.2]  (output)(display)
                                     O5  =  0    [640150.2]  (output)(display)
                                     O4  =  0    [640150.2]  (output)(display)
                                     O3  =  0    [640150.2]  (output)(display)
                                     O2  =  1    [704149.1]  (output)(display)
                                     O1  =  0    [640150.2]  (output)(display)
                                     O0  =  0    [640150.2]  (output)(display)
                                   ENX_  =  1    [35.3]  (output)(display)
                                   ENY_  =  1    [13.3]  (output)(display)
                                   ENZ_  =  0    [704133.9]  (output)(display)
                                    RS1  =  1    [384151.5]  (output)(display)
                                    SWC  =  1    [704098]  (output)(display)
  Time = 6138:1   [767125 ns]
CLK8MH --> 0     [+0/0]
  Time = 6138:2   [767187.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 6139:1   [767250 ns]
CLK8MH --> 0     [+0/0]
  Time = 6139:2   [767312.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 6140:1   [767375 ns]
CLK8MH --> 0     [+0/0]
  Time = 6140:2   [767437.5 ns]
CLK8MH --> 1     [+0/0]
```

```
  Time = 6141:1   [767500 ns]
CLK8MH --> 0     [+0/0]
  Time = 6141:2   [767562.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 6142:1   [767625 ns]
CLK8MH --> 0     [+0/0]
  Time = 6142:2   [767687.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 6143:1   [767750 ns]
CLK8MH --> 0     [+0/0]
  Time = 6143:2   [767812.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 6144:1   [767875 ns]
CLK8MH --> 0     [+0/0]
  Time = 6144:2   [767937.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 6145:1   [768000 ns]
CLK8MH --> 0     [+0/0]
  Time = 6145:2   [768062.5 ns]
CLK8MH --> 1     [+0/0]
Q2 --> 0     [+25.5/7.9]
Q1 --> 0     [+26.7/9.1]
Q0 --> 0     [+28.2/10.6]
SWC --> 0     [+33/15.4]
CNTC --> 0     [+35.6/2.6]
ENZ_ --> 1     [+41.5/7.7]
A0 --> 1     [+49.4/.5]
  Time = 6146:1   [768125 ns]
CLK8MH --> 0     [+0/0]
T2 --> 0     [+2.5/1.9]
S1 --> 0     [+3.1/19.3]
RS1 --> 0     [+9.5/7]
O2 --> 0     [+15.3/.4]
T2 --> 1     [+20.1/2.1]
RS1 --> 1     [+26.5/6.4]
  Time = 6146:2   [768187.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 6147:1   [768250 ns]
CLK8MH --> 0     [+0/0]
  Time = 6147:2   [768312.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 6148:1   [768375 ns]
CLK8MH --> 0     [+0/0]
  Time = 6148:2   [768437.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 6149:1   [768500 ns]
I5 --> 1     [+0/0]
CLK8MH --> 0     [+0/0]
  Time = 6149:2   [768562.5 ns]
CLK8MH --> 1     [+0/0]
```

```
  Time = 11257:1   [1407000 ns]
                                 I7  =  1    [0]  (input)(display)
                                 I6  =  1    [640500]  (input)(display)
                                 I5  =  1    [768500]  (input)(display)
                                 I2  =  0    [640500]  (input)(display)
                                 S0  =  1    [1280131.1]  (output)(display)
                                 S1  =  0    [1152128.1]  (output)(display)
                               LEX_  =  0    [832113.3]  (output)(display)
                               LEY_  =  0    [960113.3]  (output)(display)
                               LEZ_  =  0    [704113.3]  (output)(display)
                                LE_  =  0    [704125.6]  (output)(display)
                               CNTC  =  1    [1344100]  (output)(display)
                                 A0  =  1    [1280111.9]  (output)(display)
                                 A1  =  0    [1152115.7]  (output)(display)
                                 A2  =  1    [1152111.9]  (output)(display)
                                 A3  =  0    [15]  (output)(display)
                                 A4  =  0    [15]  (output)(display)
                                 I4  =  0    [0]  (input)(display)
                                 I3  =  0    [0]  (input)(display)
                                 I1  =  0    [0]  (input)(display)
                                 I0  =  0    [0]  (input)(display)
                                 T2  =  1    [1152145.1]  (output)(display)
                                 Q0  =  1    [1406092.6]  (output)(display)
                                 Q1  =  1    [1404091]  (output)(display)
                                 Q2  =  1    [1400089.7]  (output)(display)
                                CLR  =  1    [125]  (input)(display)
                                 RS  =  1    [125]  (input)(display)
                                OX7  =  1    [17.7]  (output)(display)
                                OY7  =  1    [17.7]  (output)(display)
                                OZ7  =  1    [17.7]  (output)(display)
                             CLK8MH  =  1    [1406937.5]  (clock)(display)
                                 O7  =  0    [1152177.1]  (output)(display)
                                 O6  =  0    [1024150.2]  (output)(display)
                                 O5  =  0    [1024150.2]  (output)(display)
                                 O4  =  0    [1024150.2]  (output)(display)
                                 O3  =  0    [1024150.2]  (output)(display)
                                 O2  =  0    [1024150.2]  (output)(display)
                                 O1  =  0    [1024150.2]  (output)(display)
                                 O0  =  0    [1024150.2]  (output)(display)
                               ENX_  =  1    [1280104]  (output)(display)
                               ENY_  =  0    [1344107.1]  (output)(display)
                               ENZ_  =  1    [1152104]  (output)(display)
                                RS1  =  1    [1152151.5]  (output)(display)
                                SWC  =  1    [1344098]  (output)(display)
  Time = 11257:1   [1407000 ns]
CLK8MH --> 0     [+0/0]
  Time = 11257:2   [1407062.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 11258:1   [1407125 ns]
CLK8MH --> 0     [+0/0]
  Time = 11258:2   [1407187.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 11259:1   [1407250 ns]
CLK8MH --> 0     [+0/0]
```

```
  Time = 11259:2   [1407312.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 11260:1   [1407375 ns]
CLK8MH --> 0     [+0/0]
  Time = 11260:2   [1407437.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 11261:1   [1407500 ns]
CLK8MH --> 0     [+0/0]
  Time = 11261:2   [1407562.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 11262:1   [1407625 ns]
CLK8MH --> 0     [+0/0]
  Time = 11262:2   [1407687.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 11263:1   [1407750 ns]
CLK8MH --> 0     [+0/0]
  Time = 11263:2   [1407812.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 11264:1   [1407875 ns]
CLK8MH --> 0     [+0/0]
  Time = 11264:2   [1407937.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 11265:1   [1408000 ns]
CLK8MH --> 0     [+0/0]
  Time = 11265:2   [1408062.5 ns]
CLK8MH --> 1     [+0/0]
Q2 --> 0     [+25.5/7.9]
Q1 --> 0     [+26.7/9.1]
Q0 --> 0     [+28.2/10.6]
SWC --> 0     [+33/15.4]
CNTC --> 0     [+35.6/2.6]
ENY_ --> 1     [+41.5/7.7]
A1 --> 1     [+49.4/.5]
A0 --> 0     [+53.2/.7]
S1 --> 1     [+59.7/17.4]
  Time = 11266:1   [1408125 ns]
CLK8MH --> 0     [+0/0]
O7 --> u     [+8.7/0]
O3 --> u     [+8.7/0]
O2 --> u     [+8.7/0]
O1 --> u     [+8.7/0]
O0 --> u     [+8.7/0]
O4 --> u     [+8.7/0]
O5 --> u     [+8.7/0]
O6 --> u     [+8.7/0]
S0 --> 0     [+13/29.2]
O0 --> 1     [+24.5/0]
O3 --> 1     [+24.5/0]
O1 --> 1     [+24.5/0]
O7 --> 0     [+25.2/0]
O2 --> 0     [+25.2/0]
O4 --> 0     [+25.2/0]
O5 --> 0     [+25.2/0]
O6 --> 0     [+25.2/0]
```

```
  Time = 11266:2   [1408187.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 11267:1   [1408250 ns]
CLK8MH --> 0     [+0/0]
  Time = 11267:2   [1408312.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 11268:1   [1408375 ns]
CLK8MH --> 0     [+0/0]
  Time = 11268:2   [1408437.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 11269:1   [1408500 ns]
CLK8MH --> 0     [+0/0]
  Time = 11269:2   [1408562.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 11270:1   [1408625 ns]
CLK8MH --> 0     [+0/0]
  Time = 11270:2   [1408687.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 11271:1   [1408750 ns]
I6 --> 0     [+0/0]
I5 --> 0     [+0/0]
I2 --> 1     [+0/0]
CLK8MH --> 0     [+0/0]
  Time = 11271:2   [1408812.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 11272:1   [1408875 ns]
CLK8MH --> 0     [+0/0]
  Time = 11272:2   [1408937.5 ns]
CLK8MH --> 1     [+0/0]
```

```
  Time = 12288:1   [1535875 ns]
                                  I7   =  1    [0]  (input)(display)
                                  I6   =  0    [1408750]  (input)(display)
                                  I5   =  0    [1408750]  (input)(display)
                                  I2   =  1    [1408750]  (input)(display)
                                  S0   =  0    [1408138]  (output)(display)
                                  S1   =  1    [1408122.2]  (output)(display)
                                LEX_   =  0    [832113.3]  (output)(display)
                                LEY_   =  0    [960113.3]  (output)(display)
                                LEZ_   =  1    [1472112.1]  (output)(display)
                                 LE_   =  0    [704125.6]  (output)(display)
                                CNTC   =  1    [1472100]  (output)(display)
                                  A0   =  0    [1408115.7]  (output)(display)
                                  A1   =  1    [1408111.9]  (output)(display)
                                  A2   =  1    [1152111.9]  (output)(display)
                                  A3   =  0    [15]  (output)(display)
                                  A4   =  0    [15]  (output)(display)
                                  I4   =  0    [0]  (input)(display)
                                  I3   =  0    [0]  (input)(display)
                                  I1   =  0    [0]  (input)(display)
                                  I0   =  0    [0]  (input)(display)
                                  T2   =  1    [1152145.1]  (output)(display)
                                  Q0   =  1    [1534092.6]  (output)(display)
                                  Q1   =  1    [1532091]  (output)(display)
                                  Q2   =  1    [1528089.7]  (output)(display)
                                 CLR   =  1    [125]  (input)(display)
                                  RS   =  1    [125]  (input)(display)
                                 OX7   =  1    [17.7]  (output)(display)
                                 OY7   =  1    [17.7]  (output)(display)
                                 OZ7   =  1    [17.7]  (output)(display)
                              CLK8MH   =  1    [1535812.5]  (clock)(display)
                                  O7   =  0    [1408150.2]  (output)(display)
                                  O6   =  0    [1408150.2]  (output)(display)
                                  O5   =  0    [1408150.2]  (output)(display)
                                  O4   =  0    [1408150.2]  (output)(display)
                                  O3   =  1    [1408149.5]  (output)(display)
                                  O2   =  1    [1472126]  (output)(display)
                                  O1   =  0    [1472125.5]  (output)(display)
                                  O0   =  0    [1472127]  (output)(display)
                                ENX_   =  1    [1280104]  (output)(display)
                                ENY_   =  1    [1408104]  (output)(display)
                                ENZ_   =  0    [1472107.1]  (output)(display)
                                 RS1   =  1    [1152151.5]  (output)(display)
                                 SWC   =  1    [1472098]  (output)(display)
  Time = 12288:1   [1535875 ns]
CLK8MH --> 0     [+0/0]
  Time = 12288:2   [1535937.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 12289:1   [1536000 ns]
CLK8MH --> 0     [+0/0]
  Time = 12289:2   [1536062.5 ns]
CLK8MH --> 1     [+0/0]
Q2 --> 0     [+25.5/7.9]
Q1 --> 0     [+26.7/9.1]
Q0 --> 0     [+28.2/10.6]
SWC --> 0     [+33/15.4]
```

```
CNTC --> 0      [+35.6/2.6]
ENZ_  --> 1     [+41.5/7.7]
A0 --> 1     [+49.4/.5]
  Time = 12290:1    [1536125 ns]
CLK8MH --> 0      [+0/0]
T2 --> 0     [+2.5/1.9]
S1 --> 0     [+3.1/19.3]
RS1 --> 0     [+9.5/7]
O2 --> 0     [+15.3/.4]
O3 --> 0     [+15.3/.4]
T2 --> 1     [+20.1/2.1]
RS1 --> 1     [+26.5/6.4]
O7 --> 1     [+51.6/.5]
O6 --> 1     [+51.6/.5]
O5 --> 1     [+51.6/.5]
  Time = 12290:2    [1536187.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 12291:1    [1536250 ns]
CLK8MH --> 0     [+0/0]
  Time = 12291:2    [1536312.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 12292:1    [1536375 ns]
CLK8MH --> 0     [+0/0]
  Time = 12292:2    [1536437.5 ns]
CLK8MH --> 1     [+0/0]
```

```
  Time = 16379:1   [2047250 ns]
                            I7  =  1    [0]  (input)(display)
                            I6  =  0    [1408750]  (input)(display)
                            I5  =  0    [1408750]  (input)(display)
                            I2  =  1    [1408750]  (input)(display)
                            S0  =  0    [1792138]  (output)(display)
                            S1  =  0    [1920128.1]  (output)(display)
                          LEX_  =  1    [1600112.1]  (output)(display)
                          LEY_  =  1    [1728112.1]  (output)(display)
                          LEZ_  =  1    [1472112.1]  (output)(display)
                           LE_  =  1    [1920169.8]  (output)(display)
                          CNTC  =  0    [1920098.1]  (output)(display)
                            A0  =  0    [1920115.7]  (output)(display)
                            A1  =  1    [1920111.9]  (output)(display)
                            A2  =  0    [1664115.7]  (output)(display)
                            A3  =  1    [1664111.9]  (output)(display)
                            A4  =  0    [15]  (output)(display)
                            I4  =  0    [0]  (input)(display)
                            I3  =  0    [0]  (input)(display)
                            I1  =  0    [0]  (input)(display)
                            I0  =  0    [0]  (input)(display)
                            T2  =  1    [1920145.1]  (output)(display)
                            Q0  =  1    [2046092.6]  (output)(display)
                            Q1  =  1    [2044091]  (output)(display)
                            Q2  =  1    [2040089.7]  (output)(display)
                           CLR  =  1    [125]  (input)(display)
                            RS  =  1    [125]  (input)(display)
                           OX7  =  1    [17.7]  (output)(display)
                           OY7  =  1    [17.7]  (output)(display)
                           OZ7  =  1    [17.7]  (output)(display)
                        CLK8MH  =  1    [2047187.5]  (clock)(display)
                            O7  =  1    [1792149.5]  (output)(display)
                            O6  =  1    [1920139.6]  (output)(display)
                            O5  =  1    [1920139.6]  (output)(display)
                            O4  =  0    [1792150.2]  (output)(display)
                            O3  =  0    [1792150.2]  (output)(display)
                            O2  =  0    [1920140.3]  (output)(display)
                            O1  =  0    [1792150.2]  (output)(display)
                            O0  =  0    [1792150.2]  (output)(display)
                          ENX_  =  1    [1664104]  (output)(display)
                          ENY_  =  1    [1792104]  (output)(display)
                          ENZ_  =  1    [1920104]  (output)(display)
                           RS1  =  1    [1920151.5]  (output)(display)
                           SWC  =  1    [1984098]  (output)(display)
  Time = 16379:1   [2047250 ns]
CLK8MH --> 0     [+0/0]
  Time = 16379:2   [2047312.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 16380:1   [2047375 ns]
CLK8MH --> 0     [+0/0]
  Time = 16380:2   [2047437.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 16381:1   [2047500 ns]
CLK8MH --> 0     [+0/0]
```

```
  Time = 16381:2   [2047562.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 16382:1   [2047625 ns]
CLK8MH --> 0     [+0/0]
  Time = 16382:2   [2047687.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 16383:1   [2047750 ns]
CLK8MH --> 0     [+0/0]
  Time = 16383:2   [2047812.5 ns]
CLK8MH --> 1     [+0/0]

  Time = 16384:1   [2047875 ns]
CLK8MH --> 0     [+0/0]
  Time = 16384:2   [2047937.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 16385:1   [2048000 ns]
CLK8MH --> 0     [+0/0]
  Time = 16385:2   [2048062.5 ns]
CLK8MH --> 1     [+0/0]
Q2 --> 0     [+25.5/7.9]
Q1 --> 0     [+26.7/9.1]
Q0 --> 0     [+28.2/10.6]
SWC --> 0     [+33/15.4]
  Time = 16386:1   [2048125 ns]
CLK8MH --> 0     [+0/0]
S0 --> 1     [+6.1/26.3]
  Time = 16386:2   [2048187.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 16387:1   [2048250 ns]
I6 --> 1     [+0/0]
I2 --> 0     [+0/0]
CLK8MH --> 0     [+0/0]
  Time = 16387:2   [2048312.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 16388:1   [2048375 ns]
CLK8MH --> 0     [+0/0]
  Time = 16388:2   [2048437.5 ns]
CLK8MH --> 1     [+0/0]
```

```
  Time = 18429:1    [2303500 ns]
                                   I7  =  1    [0]  (input)(display)
                                   I6  =  1    [2048250]  (input)(display)
                                   I5  =  0    [1408750]  (input)(display)
                                   I2  =  0    [2048250]  (input)(display)
                                   S0  =  0    [2176138]  (output)(display)
                                   S1  =  1    [2176122.2]  (output)(display)
                                 LEX_  =  1    [1600112.1]  (output)(display)
                                 LEY_  =  0    [2112113.3]  (output)(display)
                                 LEZ_  =  0    [2240113.3]  (output)(display)
                                  LE_  =  0    [2112125.6]  (output)(display)
                                 CNTC  =  1    [2240100]  (output)(display)
                                   A0  =  1    [2176111.9]  (output)(display)
                                   A1  =  1    [1920111.9]  (output)(display)
                                   A2  =  0    [1664115.7]  (output)(display)
                                   A3  =  1    [1664111.9]  (output)(display)
                                   A4  =  0    [15]  (output)(display)
                                   I4  =  0    [0]  (input)(display)
                                   I3  =  0    [0]  (input)(display)
                                   I1  =  0    [0]  (input)(display)
                                   I0  =  0    [0]  (input)(display)
                                   T2  =  1    [1920145.1]  (output)(display)
                                   Q0  =  1    [2302092.6]  (output)(display)
                                   Q1  =  1    [2300091]  (output)(display)
                                   Q2  =  1    [2296089.7]  (output)(display)
                                  CLR  =  1    [125]  (input)(display)
                                   RS  =  1    [125]  (input)(display)
                                  OX7  =  1    [17.7]  (output)(display)
                                  OY7  =  1    [17.7]  (output)(display)
                                  OZ7  =  1    [17.7]  (output)(display)
                               CLK8MH  =  1    [2303437.5]  (clock)(display)
                                   O7  =  1    [2176149.5]  (output)(display)
                                   O6  =  0    [2176150.2]  (output)(display)
                                   O5  =  0    [2176150.2]  (output)(display)
                                   O4  =  0    [2176150.2]  (output)(display)
                                   O3  =  0    [2176150.2]  (output)(display)
                                   O2  =  1    [2176149.5]  (output)(display)
                                   O1  =  0    [2176150.2]  (output)(display)
                                   O0  =  0    [2176150.2]  (output)(display)
                                 ENX_  =  1    [1664104]  (output)(display)
                                 ENY_  =  1    [2176104]  (output)(display)
                                 ENZ_  =  0    [2240107.1]  (output)(display)
                                  RS1  =  1    [1920151.5]  (output)(display)
                                  SWC  =  1    [2240098]  (output)(display)
  Time = 18429:1    [2303500 ns]
CLK8MH --> 0      [+0/0]
  Time = 18429:2    [2303562.5 ns]
CLK8MH --> 1      [+0/0]
  Time = 18430:1    [2303625 ns]
CLK8MH --> 0      [+0/0]
  Time = 18430:2    [2303687.5 ns]
CLK8MH --> 1      [+0/0]
  Time = 18431:1    [2303750 ns]
CLK8MH --> 0      [+0/0]
  Time = 18431:2    [2303812.5 ns]
CLK8MH --> 1      [+0/0]
```

```
  Time = 18432:1   [2303875 ns]
CLK8MH --> 0     [+0/0]
  Time = 18432:2   [2303937.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 18433:1   [2304000 ns]
CLK8MH --> 0     [+0/0]
  Time = 18433:2   [2304062.5 ns]
CLK8MH --> 1     [+0/0]
Q2 --> 0     [+25.5/7.9]
Q1 --> 0     [+26.7/9.1]
Q0 --> 0     [+28.2/10.6]
SWC --> 0     [+33/15.4]
CNTC --> 0     [+35.6/2.6]
ENZ_ --> 1     [+41.5/7.7]
A2 --> 1     [+49.4/.5]
A1 --> 0     [+53.2/.7]
A0 --> 0     [+53.2/.7]
  Time = 18434:1   [2304125 ns]
CLK8MH --> 0     [+0/0]
T2 --> 0     [+2.5/1.9]
S1 --> 0     [+3.1/19.3]
RS1 --> 0     [+9.5/7]
O6 --> 1     [+14.6/.5]
O5 --> 1     [+14.6/.5]
O2 --> 0     [+15.3/.4]
T2 --> 1     [+20.1/2.1]
RS1 --> 1     [+26.5/6.4]
O6 --> 0     [+57.9/.4]
O5 --> 0     [+57.9/.4]
  Time = 18434:2   [2304187.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 18435:1   [2304250 ns]
CLK8MH --> 0     [+0/0]
  Time = 18435:2   [2304312.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 18436:1   [2304375 ns]
I6 --> 0     [+0/0]
I5 --> 1     [+0/0]
CLK8MH --> 0     [+0/0]
  Time = 18436:2   [2304437.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 18437:1   [2304500 ns]
CLK8MH --> 0     [+0/0]
  Time = 18437:2   [2304562.5 ns]
CLK8MH --> 1     [+0/0]
```

```
  Time = 21507:1   [2688250 ns]
                                I7  =  1     [0]  (input)(display)
                                I6  =  0     [2688125]  (input)(display)
                                I5  =  1     [2304375]  (input)(display)
                                I2  =  0     [2048250]  (input)(display)
                                S0  =  0     [2560138]  (output)(display)
                                S1  =  0     [2688128.1]  (output)(display)
                              LEX_  =  0     [2368113.3]  (output)(display)
                              LEY_  =  0     [2112113.3]  (output)(display)
                              LEZ_  =  0     [2240113.3]  (output)(display)
                               LE_  =  0     [2112125.6]  (output)(display)
                              CNTC  =  0     [2688098.1]  (output)(display)
                                A0  =  1     [2688111.9]  (output)(display)
                                A1  =  1     [2560111.9]  (output)(display)
                                A2  =  1     [2304111.9]  (output)(display)
                                A3  =  1     [1664111.9]  (output)(display)
                                A4  =  0     [15]  (output)(display)
                                I4  =  0     [0]  (input)(display)
                                I3  =  0     [0]  (input)(display)
                                I1  =  0     [0]  (input)(display)
                                I0  =  0     [0]  (input)(display)
                                T2  =  1     [2688145.1]  (output)(display)
                                Q0  =  0     [2688090.7]  (output)(display)
                                Q1  =  0     [2688089.2]  (output)(display)
                                Q2  =  0     [2688088]  (output)(display)
                               CLR  =  1     [125]  (input)(display)
                                RS  =  1     [125]  (input)(display)
                               OX7  =  1     [17.7]  (output)(display)
                               OY7  =  1     [17.7]  (output)(display)
                               OZ7  =  1     [17.7]  (output)(display)
                            CLK8MH  =  1     [2688187.5]  (clock)(display)
                                O7  =  0     [2688177.1]  (output)(display)
                                O6  =  0     [2560150.2]  (output)(display)
                                O5  =  0     [2560150.2]  (output)(display)
                                O4  =  0     [2560150.2]  (output)(display)
                                O3  =  0     [2560150.2]  (output)(display)
                                O2  =  0     [2560150.2]  (output)(display)
                                O1  =  0     [2560150.2]  (output)(display)
                                O0  =  0     [2560150.2]  (output)(display)
                              ENX_  =  1     [2432104]  (output)(display)
                              ENY_  =  1     [2560104]  (output)(display)
                              ENZ_  =  1     [2688104]  (output)(display)
                               RS1  =  1     [2688151.5]  (output)(display)
                               SWC  =  0     [2688095.5]  (output)(display)
  Time = 21507:1   [2688250 ns]
CLK8MH --> 0     [+0/0]
  Time = 21507:2   [2688312.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 21508:1   [2688375 ns]
I6 --> 1     [+0/0]
CLK8MH --> 0     [+0/0]
  Time = 21508:2   [2688437.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 21509:1   [2688500 ns]
CLK8MH --> 0     [+0/0]
```

```
  Time = 21509:2    [2688562.5 ns]
CLK8MH --> 1      [+0/0]
  Time = 21510:1    [2688625 ns]
CLK8MH --> 0      [+0/0]
  Time = 21510:2    [2688687.5 ns]
CLK8MH --> 1      [+0/0]
  Time = 21511:1    [2688750 ns]
CLK8MH --> 0      [+0/0]
  Time = 21511:2    [2688812.5 ns]
CLK8MH --> 1      [+0/0]
```

```
   Time = 24573:1    [3071500 ns]
                                 I7  =  1     [0]  (input)(display)
                                 I6  =  1     [2688375]  (input)(display)
                                 I5  =  1     [2304375]  (input)(display)
                                 I2  =  0     [2048250]  (input)(display)
                                 S0  =  0     [2944138]  (output)(display)
                                 S1  =  1     [2944122.2]  (output)(display)
                               LEX_  =  0     [2368113.3]  (output)(display)
                               LEY_  =  0     [2112113.3]  (output)(display)
                               LEZ_  =  0     [2240113.3]  (output)(display)
                                LE_  =  0     [2112125.6]  (output)(display)
                               CNTC  =  1     [3008100]  (output)(display)
                                 A0  =  1     [2944111.9]  (output)(display)
                                 A1  =  0     [2816115.7]  (output)(display)
                                 A2  =  0     [2816115.7]  (output)(display)
                                 A3  =  0     [2816115.7]  (output)(display)
                                 A4  =  1     [2816111.9]  (output)(display)
                                 I4  =  0     [0]  (input)(display)
                                 I3  =  0     [0]  (input)(display)
                                 I1  =  0     [0]  (input)(display)
                                 I0  =  0     [0]  (input)(display)
                                 T2  =  1     [2688145.1]  (output)(display)
                                 Q0  =  1     [3070092.6]  (output)(display)
                                 Q1  =  1     [3068091]  (output)(display)
                                 Q2  =  1     [3064089.7]  (output)(display)
                                CLR  =  1     [125]  (input)(display)
                                 RS  =  1     [125]  (input)(display)
                                OX7  =  1     [17.7]  (output)(display)
                                OY7  =  1     [17.7]  (output)(display)
                                OZ7  =  1     [17.7]  (output)(display)
                             CLK8MH  =  1     [3071437.5]  (clock)(display)
                                 O7  =  0     [2944150.2]  (output)(display)
                                 O6  =  0     [2944150.2]  (output)(display)
                                 O5  =  0     [2944150.2]  (output)(display)
                                 O4  =  1     [2944149.5]  (output)(display)
                                 O3  =  1     [3008124.9]  (output)(display)
                                 O2  =  0     [3008124.4]  (output)(display)
                                 O1  =  0     [3008125.5]  (output)(display)
                                 O0  =  0     [3008127]  (output)(display)
                               ENX_  =  1     [2816104]  (output)(display)
                               ENY_  =  1     [2944104]  (output)(display)
                               ENZ_  =  0     [3008107.1]  (output)(display)
                                RS1  =  1     [2688151.5]  (output)(display)
                                SWC  =  1     [3008098]  (output)(display)
   Time = 24573:1    [3071500 ns]
 CLK8MH --> 0      [+0/0]
   Time = 24573:2    [3071562.5 ns]
 CLK8MH --> 1      [+0/0]
   Time = 24574:1    [3071625 ns]
 CLK8MH --> 0      [+0/0]
   Time = 24574:2    [3071687.5 ns]
 CLK8MH --> 1      [+0/0]
   Time = 24575:1    [3071750 ns]
 CLK8MH --> 0      [+0/0]
   Time = 24575:2    [3071812.5 ns]
 CLK8MH --> 1      [+0/0]
```

```
  Time = 24576:1   [3071875 ns]
CLK8MH --> 0     [+0/0]
  Time = 24576:2   [3071937.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 24577:1   [3072000 ns]
CLK8MH --> 0     [+0/0]
  Time = 24577:2   [3072062.5 ns]
CLK8MH --> 1     [+0/0]
Q2 --> 0     [+25.5/7.9]
Q1 --> 0     [+26.7/9.1]
Q0 --> 0     [+28.2/10.6]
SWC --> 0     [+33/15.4]
CNTC --> 0     [+35.6/2.6]
ENZ_ --> 1     [+41.5/7.7]
A1 --> 1     [+49.4/.5]
A0 --> 0     [+53.2/.7]
  Time = 24578:1   [3072125 ns]
CLK8MH --> 0     [+0/0]
T2 --> 0     [+2.5/1.9]
S1 --> 0     [+3.1/19.3]
RS1 --> 0     [+9.5/7]
O4 --> 0     [+15.3/.4]
O3 --> 0     [+15.3/.4]
T2 --> 1     [+20.1/2.1]
RS1 --> 1     [+26.5/6.4]
O7 --> 1     [+51.6/.5]
O5 --> 1     [+51.6/.5]
  Time = 24578:2   [3072187.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 24579:1   [3072250 ns]
CLK8MH --> 0     [+0/0]
  Time = 24579:2   [3072312.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 24580:1   [3072375 ns]
I5 --> 0     [+0/0]
CLK8MH --> 0     [+0/0]
  Time = 24580:2   [3072437.5 ns]
CLK8MH --> 1     [+0/0]
  Time = 24581:1   [3072500 ns]
CLK8MH --> 0     [+0/0]
  Time = 24581:2   [3072562.5 ns]
CLK8MH --> 1     [+0/0]
```

```
   Time = 27645:1   [3455500 ns]
                             I7  =  1    [0]  (input)(display)
                             I6  =  1    [2688375]  (input)(display)
                             I5  =  0    [3072375]  (input)(display)
                             I2  =  0    [2048250]  (input)(display)
                             S0  =  0    [3328138]  (output)(display)
                             S1  =  1    [3328122.2]  (output)(display)
                           LEX_  =  0    [2368113.3]  (output)(display)
                           LEY_  =  0    [2112113.3]  (output)(display)
                           LEZ_  =  0    [2240113.3]  (output)(display)
                            LE_  =  0    [2112125.6]  (output)(display)
                           CNTC  =  1    [3392100]  (output)(display)
                             A0  =  0    [3328115.7]  (output)(display)
                             A1  =  0    [3328115.7]  (output)(display)
                             A2  =  1    [3328111.9]  (output)(display)
                             A3  =  0    [2816115.7]  (output)(display)
                             A4  =  1    [2816111.9]  (output)(display)
                             I4  =  0    [0]  (input)(display)
                             I3  =  0    [0]  (input)(display)
                             I1  =  0    [0]  (input)(display)
                             I0  =  0    [0]  (input)(display)
                             T2  =  1    [3072145.1]  (output)(display)
                             Q0  =  1    [3454092.6]  (output)(display)
                             Q1  =  1    [3452091]  (output)(display)
                             Q2  =  1    [3448089.7]  (output)(display)
                            CLR  =  1    [125]  (input)(display)
                             RS  =  1    [125]  (input)(display)
                            OX7  =  1    [17.7]  (output)(display)
                            OY7  =  1    [17.7]  (output)(display)
                            OZ7  =  1    [17.7]  (output)(display)
                         CLK8MH  =  1    [3455437.5]  (clock)(display)
                             O7  =  1    [3328149.5]  (output)(display)
                             O6  =  1    [3392122.3]  (output)(display)
                             O5  =  1    [3328149.5]  (output)(display)
                             O4  =  0    [3328150.2]  (output)(display)
                             O3  =  0    [3328150.2]  (output)(display)
                             O2  =  0    [3328150.2]  (output)(display)
                             O1  =  0    [3328150.2]  (output)(display)
                             O0  =  0    [3328150.2]  (output)(display)
                           ENX_  =  1    [3200104]  (output)(display)
                           ENY_  =  1    [3328104]  (output)(display)
                           ENZ_  =  0    [3392107.1]  (output)(display)
                            RS1  =  1    [3072151.5]  (output)(display)
                            SWC  =  1    [3392098]  (output)(display)
   Time = 27645:1   [3455500 ns]
 CLK8MH --> 0      [+0/0]
   Time = 27645:2   [3455562.5 ns]
 CLK8MH --> 1      [+0/0]
   Time = 27646:1   [3455625 ns]
 CLK8MH --> 0      [+0/0]
   Time = 27646:2   [3455687.5 ns]
 CLK8MH --> 1      [+0/0]
   Time = 27647:1   [3455750 ns]
 CLK8MH --> 0      [+0/0]
   Time = 27647:2   [3455812.5 ns]
 CLK8MH --> 1      [+0/0]
```

```
  Time = 27648:1    [3455875 ns]
CLK8MH --> 0      [+0/0]
  Time = 27648:2    [3455937.5 ns]
CLK8MH --> 1      [+0/0]
  Time = 27649:1    [3456000 ns]
CLK8MH --> 0      [+0/0]
  Time = 27649:2    [3456062.5 ns]
CLK8MH --> 1      [+0/0]
Q2 --> 0      [+25.5/7.9]
Q1 --> 0      [+26.7/9.1]
Q0 --> 0      [+28.2/10.6]
SWC --> 0      [+33/15.4]
CNTC --> 0      [+35.6/2.6]
ENZ_ --> 1      [+41.5/7.7]
A0 --> 1      [+49.4/.5]
  Time = 27650:1    [3456125 ns]
CLK8MH --> 0      [+0/0]
T2 --> 0      [+2.5/1.9]
S1 --> 0      [+3.1/19.3]
RS1 --> 0      [+9.5/7]
T2 --> 1      [+20.1/2.1]
RS1 --> 1      [+26.5/6.4]
  Time = 27650:2    [3456187.5 ns]
CLK8MH --> 1      [+0/0]
  Time = 27651:1    [3456250 ns]
CLK8MH --> 0      [+0/0]
  Time = 27651:2    [3456312.5 ns]
CLK8MH --> 1      [+0/0]
  Time = 27652:1    [3456375 ns]
I6 --> 0      [+0/0]
I2 --> 1      [+0/0]
CLK8MH --> 0      [+0/0]
  Time = 27652:2    [3456437.5 ns]
CLK8MH --> 1      [+0/0]
  Time = 27653:1    [3456500 ns]
CLK8MH --> 0      [+0/0]
  Time = 27653:2    [3456562.5 ns]
CLK8MH --> 1      [+0/0]
```

```
   Time = 30717:1   [3839500 ns]
                                  I7  =  1    [0]  (input)(display)
                                  I6  =  0    [3456375]  (input)(display)
                                  I5  =  0    [3072375]  (input)(display)
                                  I2  =  1    [3456375]  (input)(display)
                                  S0  =  0    [3712138]  (output)(display)
                                  S1  =  1    [3712122.2]  (output)(display)
                                LEX_  =  1    [3520112.1]  (output)(display)
                                LEY_  =  1    [3648112.1]  (output)(display)
                                LEZ_  =  1    [3776112.1]  (output)(display)
                                 LE_  =  0    [2112125.6]  (output)(display)
                                CNTC  =  1    [3776100]  (output)(display)
                                  A0  =  1    [3712111.9]  (output)(display)
                                  A1  =  1    [3584111.9]  (output)(display)
                                  A2  =  1    [3328111.9]  (output)(display)
                                  A3  =  0    [2816115.7]  (output)(display)
                                  A4  =  1    [2816111.9]  (output)(display)
                                  I4  =  0    [0]  (input)(display)
                                  I3  =  0    [0]  (input)(display)
                                  I1  =  0    [0]  (input)(display)
                                  I0  =  0    [0]  (input)(display)
                                  T2  =  1    [3456145.1]  (output)(display)
                                  Q0  =  1    [3838092.6]  (output)(display)
                                  Q1  =  1    [3836091]  (output)(display)
                                  Q2  =  1    [3832089.7]  (output)(display)
                                 CLR  =  1    [125]  (input)(display)
                                  RS  =  1    [125]  (input)(display)
                                 OX7  =  1    [17.7]  (output)(display)
                                 OY7  =  1    [17.7]  (output)(display)
                                 OZ7  =  1    [17.7]  (output)(display)
                              CLK8MH  =  1    [3839437.5]  (clock)(display)
                                  O7  =  1    [3712149.5]  (output)(display)
                                  O6  =  1    [3712149.5]  (output)(display)
                                  O5  =  0    [3776123.7]  (output)(display)
                                  O4  =  0    [3712150.2]  (output)(display)
                                  O3  =  0    [3712150.2]  (output)(display)
                                  O2  =  0    [3712150.2]  (output)(display)
                                  O1  =  0    [3712150.2]  (output)(display)
                                  O0  =  0    [3712150.2]  (output)(display)
                                ENX_  =  1    [3584104]  (output)(display)
                                ENY_  =  1    [3712104]  (output)(display)
                                ENZ_  =  0    [3776107.1]  (output)(display)
                                 RS1  =  1    [3456151.5]  (output)(display)
                                 SWC  =  1    [3776098]  (output)(display)
   Time = 30717:1   [3839500 ns]
 CLK8MH --> 0      [+0/0]
   Time = 30717:2   [3839562.5 ns]
 CLK8MH --> 1      [+0/0]
   Time = 30718:1   [3839625 ns]
 CLK8MH --> 0      [+0/0]
   Time = 30718:2   [3839687.5 ns]
 CLK8MH --> 1      [+0/0]
   Time = 30719:1   [3839750 ns]
 CLK8MH --> 0      [+0/0]
   Time = 30719:2   [3839812.5 ns]
 CLK8MH --> 1      [+0/0]
```

```
  Time = 30720:1    [3839875 ns]
 CLK8MH --> 0     [+0/0]
  Time = 30720:2    [3839937.5 ns]
 CLK8MH --> 1     [+0/0]
  Time = 30721:1    [3840000 ns]
 CLK8MH --> 0     [+0/0]
  Time = 30721:2    [3840062.5 ns]
 CLK8MH --> 1     [+0/0]
 Q2 --> 0      [+25.5/7.9]
 Q1 --> 0      [+26.7/9.1]
 Q0 --> 0      [+28.2/10.6]
 SWC --> 0      [+33/15.4]
 CNTC --> 0      [+35.6/2.6]
 ENZ_ --> 1      [+41.5/7.7]
 A3 --> 1      [+49.4/.5]
 A1 --> 0      [+53.2/.7]
 A2 --> 0      [+53.2/.7]
 A0 --> 0      [+53.2/.7]
  Time = 30722:1    [3840125 ns]
 CLK8MH --> 0     [+0/0]
 T2 --> 0      [+2.5/1.9]
 S1 --> 0      [+3.1/19.3]
 RS1 --> 0      [+9.5/7]
 T2 --> 1      [+20.1/2.1]
 RS1 --> 1      [+26.5/6.4]
 LE_ --> 1      [+44.8/5.1]
  Time = 30722:2    [3840187.5 ns]
 CLK8MH --> 1     [+0/0]
  Time = 30723:1    [3840250 ns]
 CLK8MH --> 0     [+0/0]
  Time = 30723:2    [3840312.5 ns]
 CLK8MH --> 1     [+0/0]
  Time = 30724:1    [3840375 ns]
 CLK8MH --> 0     [+0/0]
  Time = 30724:2    [3840437.5 ns]
 CLK8MH --> 1     [+0/0]
  Time = 30725:1    [3840500 ns]
 CLK8MH --> 0     [+0/0]
  Time = 30725:2    [3840562.5 ns]
 CLK8MH --> 1     [+0/0]
```

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] G. K. Brown, N. L. Schulte Pason, E. J. Timm, C. L. Burton and D. E. Marshall, "Apple packing line impact damage reduction," *Applied Engineering in Agriculture*, vol. 6, no. 6, pp. 789-794, 1990.

[2] S. S. Sober, H. R. Zapp and G. K. Brown, "Simulated packing line impacts for apple bruise prediction," *Transactions of the ASAE*, vol. 33, no. 2, pp. 629-636, 1990.

[3] H. R. Zapp, S. H. Ehlert, G. K. Brown, P. R. Armstrong and S. S. Sober, "Advanced instrumentation sphere (IS) for impact measurement," *Transactions of the ASAE*, vol. 33, no. 3, pp. 955-960, 1990.

[4] T. Forbush, "Surface-Mount Chip Design for Instrumented Sphere Miniaturization," in *ASAE International Winter Meeting*, Paper No. 916589, Chicago, IL, Dec. 17-20, 1991.

[5] H. Tseng, M. Aslam, H. R. Zapp, G. K. Brown and C. R. Eagen, "Multichip Housing for Instrumented Sphere Miniaturization," in *ASAE International Winter Meeting*, Paper No. 916588, Chicago, IL, Dec. 17-20, 1991.

[6] H. Tseng, H. R. Zapp, M. Aslam and G. K. Brown, "Smart Logic Design for Instrumented Sphere Miniaturization," in *ASAE International Summer Meeting*, Paper No. 926070, Charlotte, NC, Jun. 21-24, 1992.

[7] H. Tseng, H. R. Zapp, M. Aslam and G. K. Brown, "Smart Logic Design for Instrumented Sphere Miniaturization," *submitted to Transactions of the ASAE*, 1993

[8] F. J. Kurdahi and A. C. Parker, "Techniques for Area Estimation of VLSI Layouts"

[9] B. Hoppe, G. Neuendorf, D. Schmitt-Landsiedel and W. Specks, "Optimization of High-Speed CMOS Logic Circuits with Analytical Models for Signal Delay, Chip Area, and Dynamic Power Dissipation," *IEEE Transactions on Computer-Aided Design*, vol. 9, no. 3, pp. 236-247, 1990.

[10] D. Stark and M. Horowitz, "Techniques for Calculating Currents and Voltages in VLSI Power Supply Networks," *IEEE Transactions on Computer-Aided Design*, vol. 9, no. 2, pp. 126-132, 1990.

[11] S. Chowdhury and J. S. Barkatullah, "Current Estimation in MOS IC Logic Circuits," *IEEE International Conference Computer Aided Design Digest Technical Papers*, pp. 212-215, Nov., 1988.

[12] A. Tyagi, "Hercules: A Power Analyzer for MOS VLSI Circuits," *IEEE International Conference Computer Aided Design Digest Technical Papers*, pp. 530-533, 1987.

[13] C. F. Fey and D. E. Paraskevopoulos, "A Techno-Economic Assessment of Application-Specific Integrated Circuits: Current Status and Future Trends," *Proceedings of the IEEE*, vol. 75, no. 6, pp. 829-841, 1987.

[14] R. K. Spielberger, C. D. Huang, W. H. Nunne, A. H. Mones, D. L. Fett and F. L. Hampton, "Silicon-on-Silicon Packaging," *IEEE Transactions on Component, Hybrids, and Manufacturing Technology*, vol. 7, no. 2, pp. 193-196, 1984.

[15] E. T. Lewis, "The VLSI Package-An Analytical Review," *IEEE Transactions on Component, Hybrids, and Manufacturing Technology*, vol. 7, no. 2, pp. 197-201, 1984.

[16] R. R. Johnson, "Multichip modules: next-generation packages," *IEEE Spectrum*, pp. 34-48, March, 1990.

[17] T. Yamada, K. Otsuka, K. Okutani and K. Sahara, "Low Stress Design of Flip Chip Technology for Si on Si Multichip Modules," *Proceedings 5th International Electronic Packaging Conference*, pp. 551-557, 1985.

[18] R. C. Rider, "Elastic Behavior of a Pseudo Fruit for Determining Bruise Damage to Fruit During Mechanized Handling," *M.S. Thesis*, University of California, Davis, CA, 1969.

[19] R. C. Rider, R. B. Fridley and M. O'Brien, "Elastic Behavior of a Pseudo-fruit for Determining Bruise Damage to Fruit During Mechanized Handling," *Transactions of the ASAE*, vol. 16, no. 2, pp. 241-244, 1973.

[20] J. M. Pullen and R. G. Diener, "A Low-Cost FET Triaxial Accelerometer with Vector Summing Capabilities," *ASAE Paper No. 71-509*, ASAE, St. Joseph, MI, 1971.

[21] M. O'Brien, R. B. Fridley, J. R. Goss and J. Shubert, "Telemetry for Investigating Force on Fruits During Handling," *Transactions of the ASAE*, vol. 16, no. 2, pp. 245-247, 1973.

[22] W. H. Aldred and J. J. Burch, "Telemetry and Microcomputer System Aids Investigation of Forces on Peaches During Mechanical harvesting," *ASAE Paper No. 77-1527*, ASAE, St. Joseph, MI, 1977.

[23] W. H. Jenkins and E. G. Humphries, "Impact Damage Assessment Technique," *Transactions of the ASAE*, vol. 25, no. 1, pp. 54-57, 1982.

[24] J. L. Halderson, C. L. Peterson and R. C. Daigh, "A Telemetry Device for Impact Determination," *Proceedings of the National Conference on Agricultural Electronics Application*, no. 2, pp. 773-780, Chicago, Dec. 11-13, 1983.

[25] J. L. Halderson and A. Skrobacki, "Dynamic Performance of an Impact Telemetry System," *ASAE Paper No. 86-3030*, ASAE, St. Joseph, MI, 1986.

[26] G. Anderson and R. Parks, "The Electronic Potato," *Departmental Note No. SIN/390*, The British Society for Research in Agricultural Engineering, 1984.

[27] G. L. Kerr and K. I. Wilkie, "Damage Detection During the Handling of Fruits and Vegetables," *CSAE Paper No. 85-202*, CSAE, Annual Meeting, Charlottetown, Canada, June 23-27, 1985.

[28] S. Siyami, "Data Acquisition System for Study of Impact Damage to Apples," *Ph. D.*

Dissertation, Michigan State University, East Lansing, MI, 1986.

[29] D. A. Lowther, C. B. Giles and I. C. Leszkowicz, "The Design and Construction of a General Purpose Laboratory Data Acquisition System," *IEEE Transactions on Instrumentation and Measurement*, vol. 29, no. 2, pp. 116-119, 1980.

[30] J. J. Hill and W. E. Alderson, "Design of a Microprocessor-Based Digital Wattmeter," *IEEE Transactions on Industrial Electronics and Control Instrumentation*, vol. 28, no. 3, pp. 180-184, 1981.

[31] E. E. Wallingford, "A Fast, Simple Microcomputer-Controlled Data Acquisition System," *IEEE Transactions on Instrumentation and Measurement*, vol. 31, no. 2, pp. 137-139, 1982.

[32] G. Sridharan, "Microcomputer-Based Synchronous Multichannel Data Acquisition System," *IEEE Transactions on Industrial Electronics*, vol. 31, no. 4, pp. 289-291, 1984.

[33] V. Adam, M. Urbina and R. E. Suarez, "Telemetric Seismic Data-Acquisition System," *IEEE Transactions on Instrumentation and Measurement*, vol. 34, no. 1, pp. 81-84, 1985.

[34] D. A. Ahrens and S. W. Searcy, "Monitoring Grazing Activities With a Data-Logging System," *Agricultural Engineering*, vol. 66, no. 1, pp. 18-20, 1985.

[35] V. C. Negro, "A Battery Operated Bubble Memory Data-Acquisition System," *IEEE Transactions on Instrumentation and Measurement*, vol. 37, no. 2, pp. 305-308, 1988.

[36] S. S. Leung, P. D. Fisher and M. A. Shanblatt, "A Conceptual Framework for ASIC Design," *Proceedings of the IEEE*, vol. 76, no. 7, pp. 741-755, 1988.

[37] A. R. Newton and A. L. Sangiovanni-Vincentelli, "Computer-Aided Design for VLSI Circuits," *IEEE computer*, pp. 38-60, April, 1986.

[38] H. W. Carter, "Computer-Aided Design of Integrated Circuits," *IEEE computer*, pp. 19-36, April, 1986.

[39] M. R. Lightner, "Modeling and Simulation of VLSI Digital Systems," *Proceedings*

of the IEEE, vol. 75, no. 6, pp. 786-796, 1987.

[40] S. Lacobovici and C. Ng., "VLSI and System Performance Modeling," *IEEE Micro*, pp. 59-72, August, 1987.

[41] D. D. Gajski and R. H. Kuhn, "Guest Editors' Introduction: New VLSI Tools," *IEEE Computer*, pp. 11-14, December, 1983.

[42] L. Waller, "How MOSIS Will Slash the Cost of IC Prototyping," *Electronics*, pp. 48-49, March, 1986.

[43] T. M. McWilliams and Lawrence C. Widdoes, Jr., "SCALD: Structured Computer-Aided Logic Design," *Computer Science Department, Stanford University and Lawrence Livermore Laboratory*, University of California. pp. 271-277, 1978.

[44] D. K. Reinhard, *Introduction to Integrated Circuit Engineering*. Boston: Houghton Mifflin Company, 1987.

[45] R. L. Geiger, P. E. Allen and N. R. Strader, *VLSI Design Techniques for Analog and Digital Circuits*. New York: Mcgraw-Hill, Inc., 1990.

[46] C. Mead and L. Conway, *Introduction to VLSI Systems*. Reading, Massachusetts: Addison-Wesley Publishing Company, Inc., 1980.

[47] N. H. E. Weste and K. Eshraghian, *Principles of CMOS VLSI Design: A Systems Perspective*. Reading, Massachusetts: Addison-Wesley Publishing Company, Inc., 1985.

[48] D. V. Heinbuch, *CMOS3 Cell Library*. Reading, Massachusetts: Addison-Wesley Publishing Company, Inc., 1988.

[49] J. Allen, "Performance-Directed Synthesis of VLSI Systems," *Proceedings of the IEEE*, vol. 78, no. 2, pp. 336-355, 1990.

[50] M. C. Mcfarland, A. C. Parker and R. Camposano, "The High-Level Synthesis of Digital Systems," *Proceedings of the IEEE*, vol. 78, no. 2, pp. 301-318, 1990.

[51] R. Camposano and W. Rosenstiel, "Synthesizing Circuits From Behavioral Descriptions," *IEEE Transactions on Computer-Aided Design*, vol. 8, no. 2, pp. 171-180, 1989.

[52] M. Wu and I. N. Hajj, "Switching Network Logic Approach to Sequential MOS Circuit Design," *IEEE Transactions on Computer-Aided Design*, voL 8, no. 7, pp. 782-794, 1989.

[53] T. H. Meng, R. W. Brodersen and D. G. Messerschmitt, "Automatic Synthesis of Asynchronous Circuits from High-Level Specifications," *IEEE Transactions on Computer-Aided Design*, vol. 8, no. 11, pp. 1185-1205, 1989.

[54] K. Karakotsios, *CMOS Logic Databook*. pp.2-71 ~ 2-75. Santa Clara, CA: National Semiconductor, 1988.

[55] *CMOS Logic Databook*. pp.3-3 ~ 3-8. Santa Clara, CA: National Semiconductor, 1988.

[56] *CMOS Logic Databook*. pp. 2-3 ~ 2-5, pp. 2-13 ~ 2-15. Dallas, TX: Texas Instruments, 1989.

[57] *CMOS Logic Databook*. pp. 59-60, pp. 63-64. Korea: Samsung Electronics, 1990.

[58] R. C. Eden, "Applicability of Diamond Substrates to Multi-Chip Modules," *ISHM 1991 Proceedings*, pp. 363-367, 1991.

[59] P. Lux and K. Roberts, "Electronic Circuits on Diamond Substrates," *General Dynamics/Electronics Division*, San Diego, CA.

[60] E. F. Borchelt and G. Lu, "Use of CVD Diamond Substrates in Electronic Applications," *6th International SAMPE Electronics Conference*, pp. 371-380, 1992.

[61] N. B. Nguyen, "Adoption of Advanced Materials in Hybrid Packaging Techniques for Ultra High Power Semiconductor Devices" *Power Conversion Proceedings*, pp. 130-133, 1992.

[62] B. Prince and R. H. W. Salters, "ICs Going on a 3-V Diet", *IEEE Spectrum*, pp. 22-25, May, 1992.

[63] J. Williams, "Mixing 3-V and 5-V ICs", *IEEE Spectrum*, pp. 40-42, March, 1993.

[64] A. R. Newton and A. L. Sangiovanni-Vincentelli, "CAD Tools for ASIC Design," *Proceedings of the IEEE*, vol. 75, no. 6, pp. 765-776, 1987.

[65] H. N. Brady and J. Blanks, "Automatic Placement and Routing Techniques for Gate Array and Standard Cell Designs," *Proceedings of the IEEE*, vol. 75, no. 6, pp. 797-806, 1987.

[66] A. C. Parker and S. Hayati, "Automating the VLSI Design Process Using Expert Systems and Silicon Compilation," *Proceedings of the IEEE*, vol. 75, no. 6, pp. 777-785, 1987.

[67] R. K. Cavin and J. L. Hilbert, "Design of Integrated Circuits: Directions and Challenges," *Proceedings of the IEEE*, vol. 78, no. 2, pp. 418-435, 1990.

[68] D. S. Harrison, A. R. Newton, R. L. Spickelmier and T. J. Barnes, "Electronic CAD Frameworks," *Proceedings of the IEEE*, vol. 78, no. 2, pp. 393-417, 1990.

[69] H. Deman, F. Catthoor, G. Goossens, J. Vanhoof, J. V. Meerbergen, S. Note and J. Huisken, "Architecture-Driven Synthesis Techniques for VLSI Implementation of DSP Algorithm," *Proceedings of the IEEE*, vol. 78, no. 2, pp. 319-335, 1990.