PDP-X Technical Memorandum # 24

Title: PDP-X/II Register Section (Proposed)

Author(s): L. Seligman

Index Keys: Register Section
Design Decisions
Processor

Distribution Keys: A

Obsolete: None

Revision: None

Date: September 28, 1967

A proposed register organization for PDP-X/II is described below. Timing in this section of the processor is constrained by main-core memory/control memory timing relations detailed in TM #21. The logic described here is capable of performing a 16 bit addition between, for example, a word in fast memory and the AR register in the 200 ns internal cycle allowed. It includes rewrite into fast memory and provides 15 ns for control memory skew. While the main operation is taking place, the MI register may be concurrently shifted, providing for a complete multiply or divide step.

1. General Description

The register section (see Fig. 1) separates into five inter-dependent parts:

a) registers (AR, MI) and Fast Memory (FM)

These registers hold the temporary data necessary for instruction fetch, address calculation, and actual execution. A typical internal processor cycle entails combining a word from fast memory and a register and returning the result to a (possibly distinct) FM word or a register.

The Arithmetic Register (AR) typically holds partial results during internal calculations while the Memory Interface (MI) register typically receives processor generated addresses and data words for transmission to the memory system. These definitions are not comprehensive; the flow charts must be followed to determine the actual use of a given register during any particular

operation. The fast memory contains part of the state word of the active process, primarily the accumulators, index registers, and the program counter. During the execution portion of, for example, an ADD instruction the appropriate accumulator in fast memory would be fetched, the memory data word added, and the sum rewritten into fast memory.

b) input gates and latch

The input gates serve to determine the source(s) of adder input(s). Two independent gate sets are used, one to control adder inputs on the A bus, the other to control adder inputs on the B bus. Most B bus input signals can appear on the bus in either true or complemented form. The B bus typically receives from memory an instruction word or a data operand while the A bus typically receives an accumulator operand or index register from fast memory.

The latch logic permits disconnecting the input signals during a bus operation yet still retaining the proper data at the adder input terminals. This two phase clocking system is required by the fast memory which cannot simultaneously read and write.

c) conditional sum adder

The conditional sum adder, as shown in Fig. 2, is considerably faster than an ordinary 16 bit adder using similar technology (Q). The adder is divided into

four groups of 4 bits each; for each group conditional sum is calculated simultaneously under the assumptions of both carry in and no carry into the group. Concurrently, carry into each group is independently calculated and the selection signals so derived for each group are used to select one of the conditional sums for use as the actual sum. The carry out terminals of each quadruple adder package appear considerably before the sum outputs and provide the group carry propagate and group carry generate signals needed in the formation of the selection signals. Selection takes place in the selector described below. The least significant group does not require conditional sum techniques; an add enable signal is also provided.

d) selector

The selector is used to a) gate the proper conditional sum b) to perform the logical AND operation and c) to permit shifting the MI register. Thus, when the MI is shifted during a divide step, adder selection signals are gated off and the MI register data is connected to the shifter through the selector. Such connection bypasses the adder while permitting it to be concurrently forming a sum. If the AR is enabled through the selector, its zeros mask ones of the adder output forming the logical AND of the AR register and whatever data is placed on either the A or B buses.

e) shifter

The shifter provides for entering data into a register, memory, or fast memory either as it appears at the selector output, shifted one bit position right or left, or with the two bytes interchanged. When shifting right or left, the end bit is supplied as described in TM #13 for the particular operation.

2. Timing Assumptions (2)

		Max. Propagation delay (ns)	Input Setup (ns)
a)	gate	10	-
b)	flip flop	40	20
c)	quad adder	50 (sum) 25 (CRY)	-
d)	buffer	10	-
e)	fast memory	30 (access)	25 (write duration)

In addition it was assumed that 30 ns (3 gate delays) would be necessary to setup an address at the FM inputs after the decision to use either the read or write address has been made.

3. Timing Descriptions

Data flow under the above timing assumptions is shown in Fig. 3.

	Where	Time Spent (ns)	Note
a)	FM address	30	3 gate delays
b)	FM access	30	spec
c)	input gate	10	1 gate delay
d)	CS adder	50	sum spec on adder
e)	selector	10	1 gate delay
f)	shifter	10	1 gate delay
g)	write gates*	20	2 gate delays
h)	FM write	25	spec
		185	

MQSTEP Flow

	Where	Time Spent (ns)	Note
i)	selector	10	1 gate delay
j)	shifter	10	1 gate delay
k)	MI setup	20	spec
l)	MI propagation	0	overlaps rest of cycle .
		40	

*write gates - A write '0' or a write '1' signals must be brought true during the FM write period in order to store data. Two gate delays are required in the data path to create these signals.

In order for data to flow through the register section at maximum speed, appropriate control signals must be available before the data arrives. These periods are shown in Fig. 3 by solid horizontal lines. Wiggly lines indicate periods when the control signals may be in an unknown state due to uncertainty in the position of the control signals. Note that the control must be worst-case designed as is the register section since gate delays and skew (uncertainty) in this area is an appreciable fraction of the delay through the register section.

In addition, certain interlocks are necessary to ensure that input enabling signals disappear before the corresponding flip flop (or latch) data input changes. These interlocks are indicated by the downward arrows between wiggly lines and include:

a) WRITE MI must be down before the selector/shifter is changed for the main cycle operation

b) LATCH must be up before the input gates are disabled and the input gates must be disabled before the FM address can change from READ (source) to WRITE (destination)

c) WRITE FM must be down before the selector/shifter, latch, and write address may change.

Only three timing signals are required during the cycle; the required accuracy is detailed in the following section. Note that no provision has been made for skew in control memory sense flip flops. This would appear as additional read access time. Hence the 185 ns given here as the register cycle is not the complete processor internal cycle.

4. Timing Control

The major timing unit is built around a recirculating delay line with a period of 200 ns, much as it is in PDP-9. Those sections of the timing logic that are concerned with details of control memory current, etc., as well as syncronization with the rest of the processor are not detailed here.

The three pulses required for register transfer operations are To (corresponds to Control Memory word actually changing hence CM strobe occurs somewhat earlier) T_1 and T_2. In addition, the latch signal and the write signal are generated by this logic as shown in Fig. 4.

Signal	Minoccurence (ns)	Maxoccurence (ns)	
T_o	0	0	
T_1	95	105	
T_2	145	155	
L(1)	95	115	going positive
L(0)	95	125	going negative
Write (1)	145	165	going positive
Write (0)	145	175	going negative

The other signals in the system derive from the above and the following control memory word bits:

Name	# Bits	Function
MQSTEP	1	Indicates a multiply or divide step.
MD	1	Direction of shift, ie, multiply or divide.
SS	3	Selector shifter control
IGN	8	Input gate controls, IG1 → FMA
		(Left right half controls not shown IG2 → FMB IG3 → FMC

Name	# Bits	Function
ARI	1	Place result in AR
MII	1	" " MI
FMI	1	" " FM
RAS	2	FM address source during read
WAS	2	FM address source during write

The derived signals are shown in Fig. 5 where MILOAD and FMA are actual, buffered control signals into the register group. Buffered gates have a 'B' in them; FMLOAD is a bundle. The composite signal timing (below) follows from the timing diagram in Fig. 6.

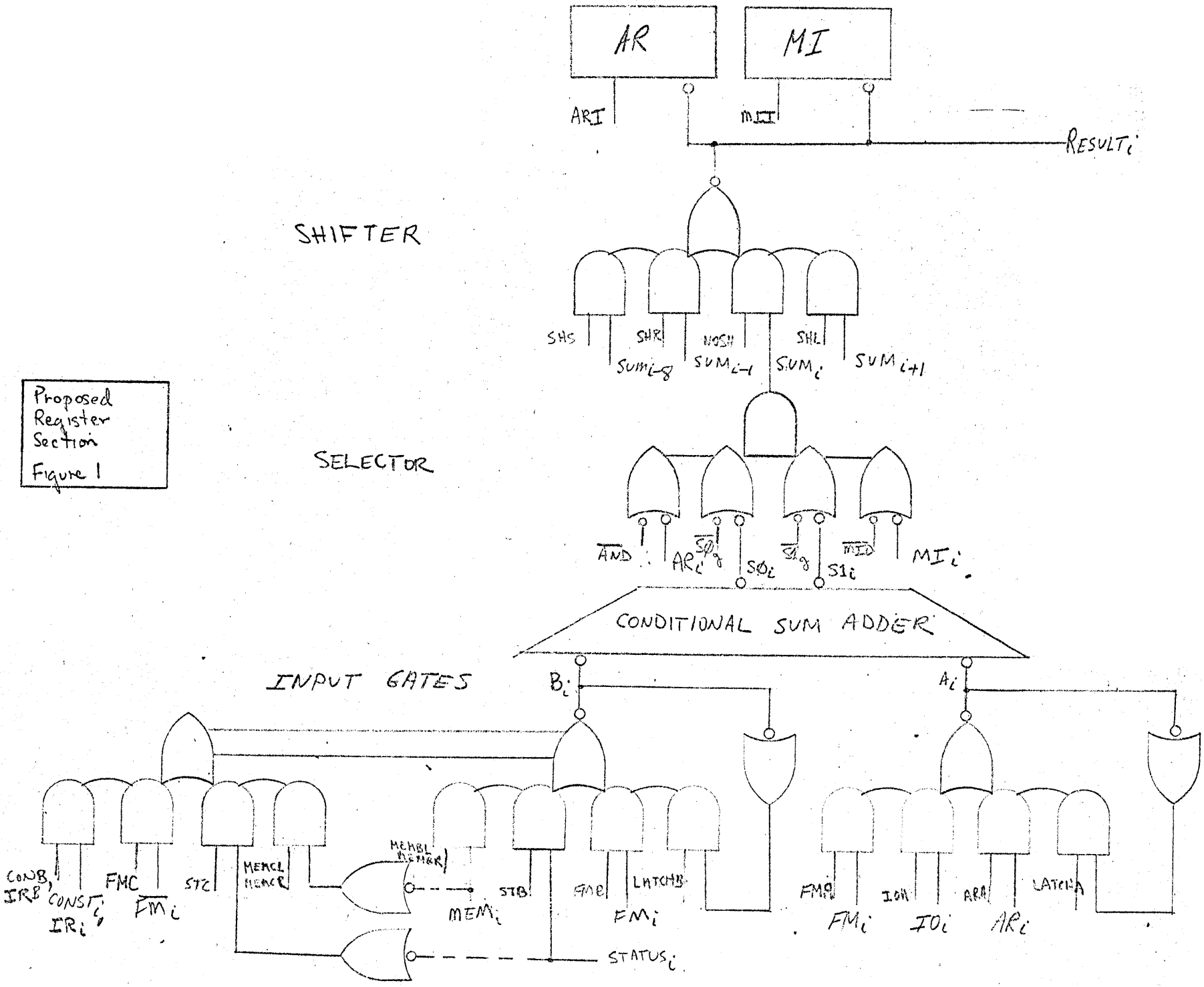
Signal	Minoccurence	Maxoccurence	Comment
FMA	95	125	one of several similar going off (negative)
DA	95	145	going off (negative)
FMLOAD	200	200	plus negligivle pulse skew
SSEN*	95	115	

*Note that MI, since it shifts, cannot need skew protection.

5. References

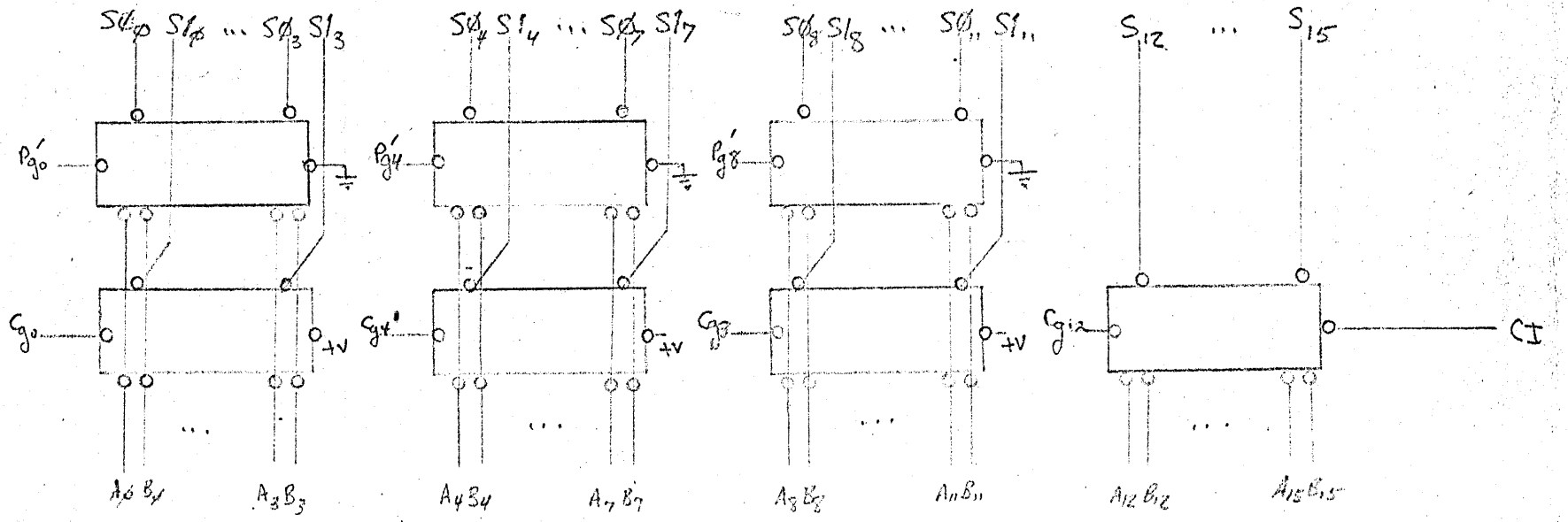
a) O. J. Bedrij, Carry-Select Adder, IRE Trans on E.C., EC-11, 340-346; June 1962.

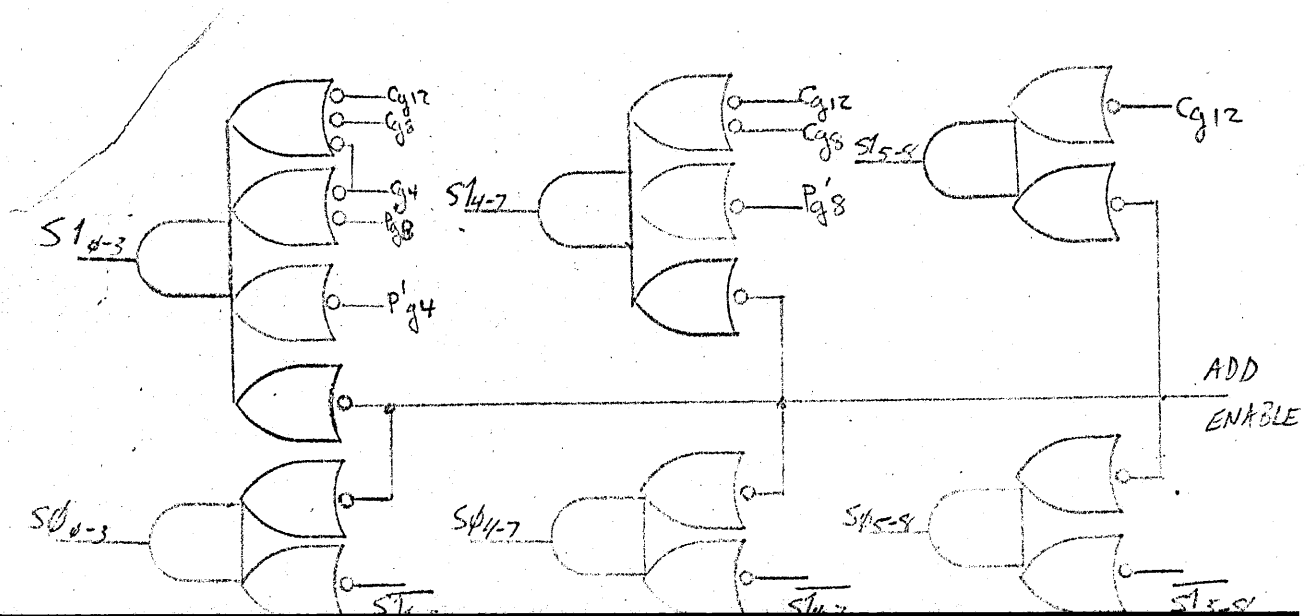
b) Texas Instruments series 74H Integrated Circuits Data Sheets.



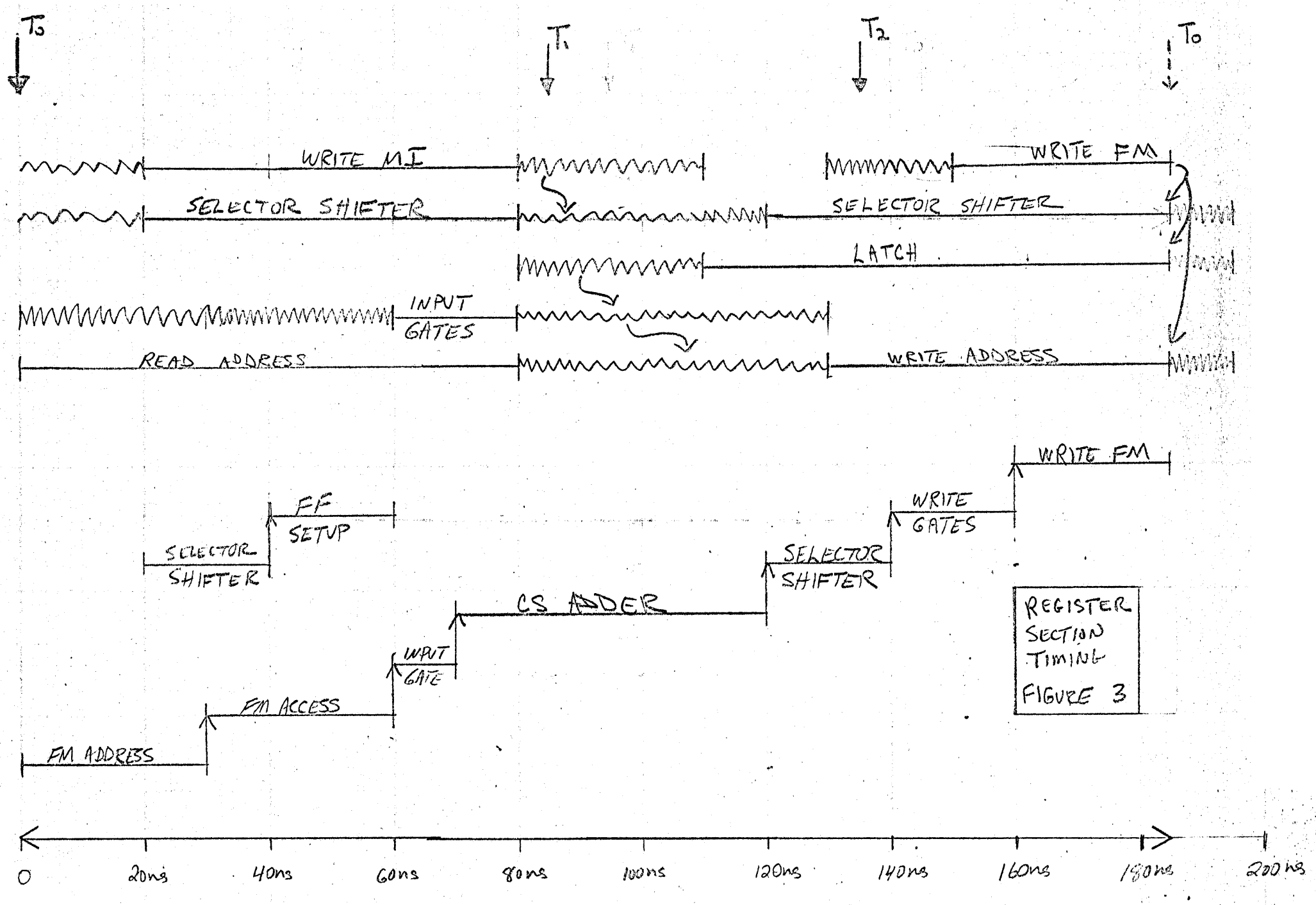
Proposed Register Section

Figure 1





Conditional Sum Adder
50ns
Figure 2


T0
T1
T2
T0
WRITE MI
WRITE FM
SELECTOR SHIFTER
SELECTOR SHIFTER
LATCH
INPUT GATES
READ ADDRESS
WRITE ADDRESS
WRITE FM
FF SETUP
WRITE GATES
SELECTOR SHIFTER
SELECTOR SHIFTER
CS ADDER
INPUT GATE
FM ACCESS
FM ADDRESS
REGISTER SECTION TIMING
FIGURE 3
0
20ns
40ns
60ns
80ns
100ns
120ns
140ns
160ns
180ns
200ns

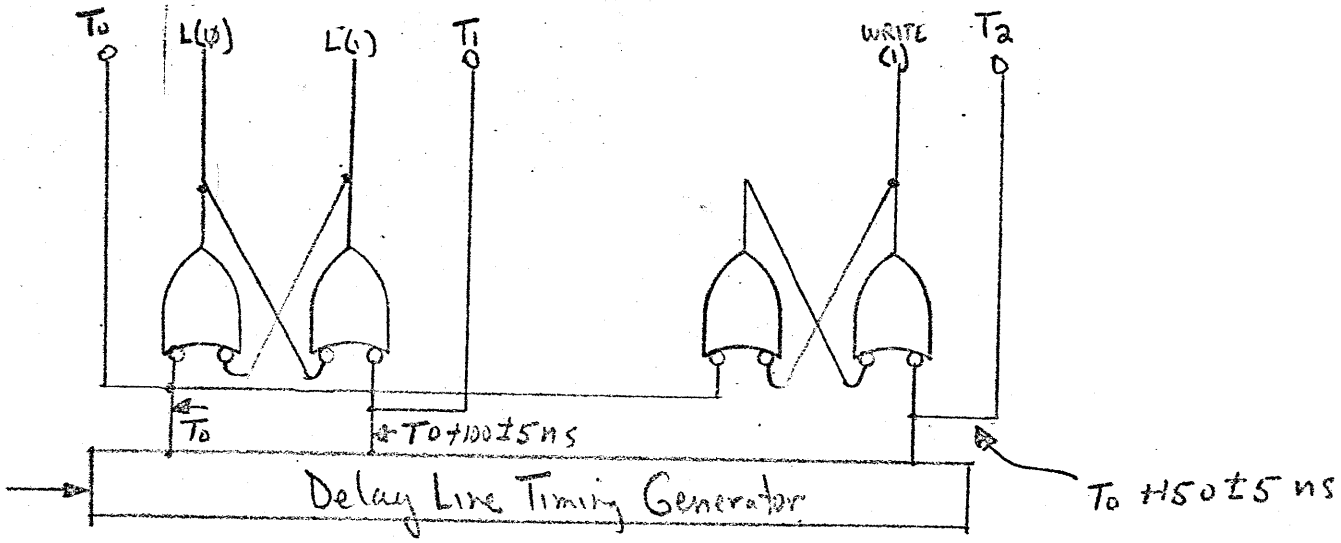
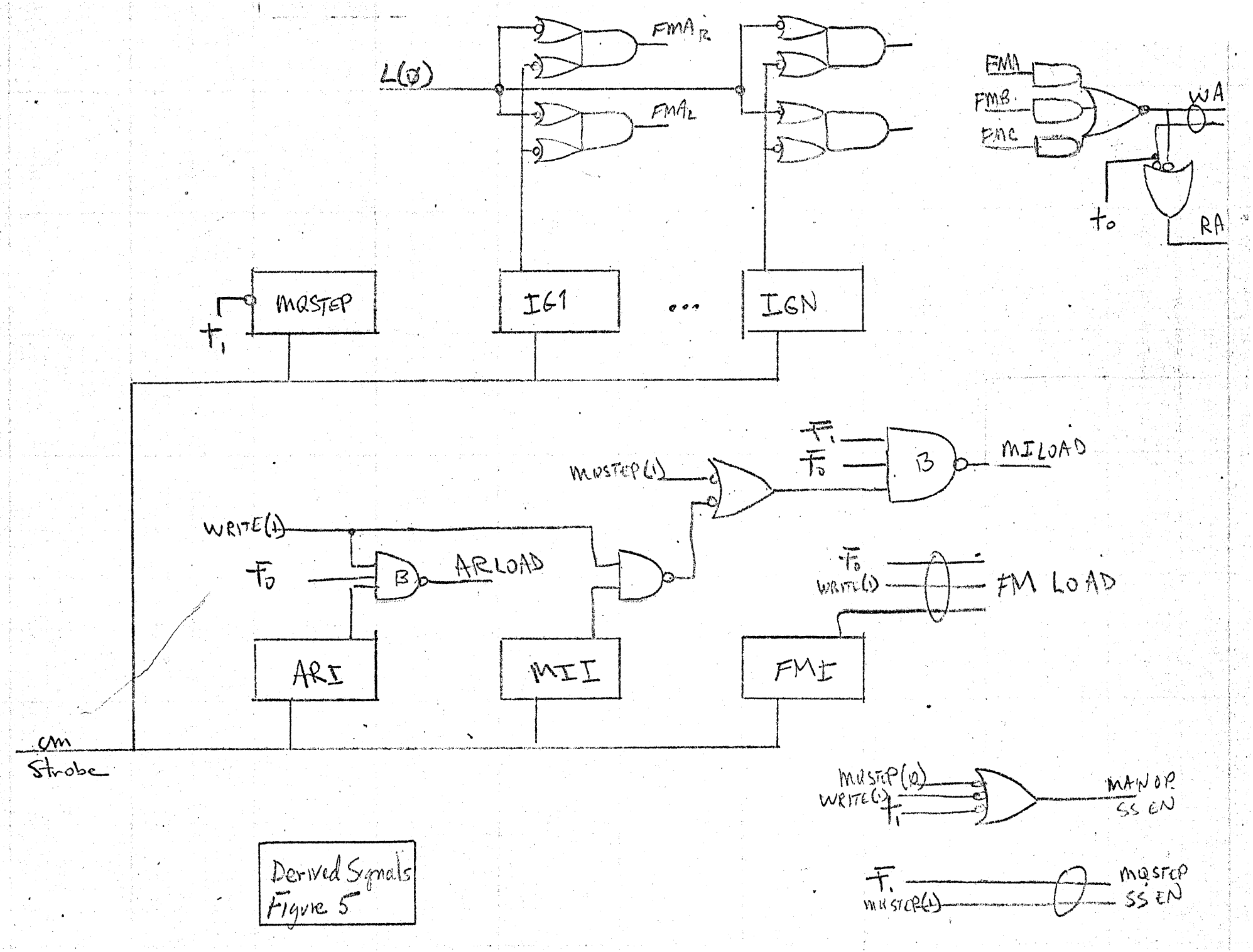

T0
L(φ)
L(ϕ)
T1
WRITE (1)
T2
T0
T0+100±5ns
Delay Line Timing Generator
T0 +50±5 ns

Figure 4



Derived Signals
Figure 5

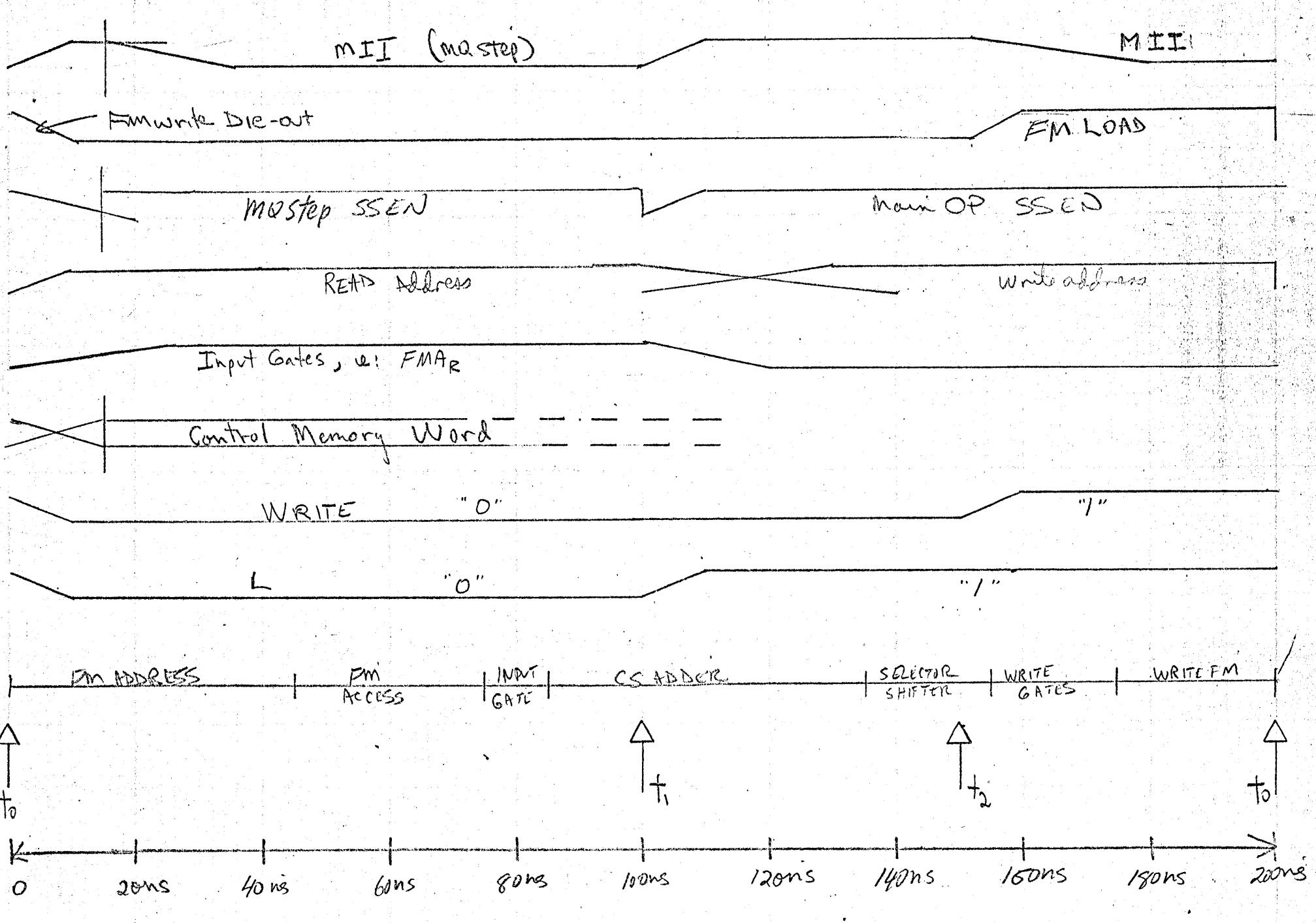


Figure 6 - DETAILED TIMING