

TTL 시뮬레이터의 콤파일러 設計에 關한 研究 (A study on the design of the compiler for a TTL simulator)

慎 哲 宰* · 金 容 得**

(Shin, Chull Chai and Kim, Yong Deak)

要 約

本 論文에서는 特殊 目的의 小型 電子計算機를 單一 배스線 方法에 의한 TTL 集積回路로 設計하였 으며, 各 指令語의 長이를 16비트로 構成함으로써 더욱 簡便한 콤파일러를 研究하였다. 또한 本 實驗에서 160nsec의 基本 週期를 擇함으로써, TTL IC의 動作時間에 따른 最適値와 記憶素 子の 액세스 타입이 같게 되었다고, 構成 回路도 간단하게 만들어졌으며, 모든 프로그램도 만족하게 처리 되었다.

Abstract

The special mini-computer was designed with the one-bus line systems employing the integrated circuits, and was studied by the method of easily making the compiler in 16 bits with each instruction fields.

When the 160 nano seconds for a fundamental cycle were used, the optimum operating time for a TTL IC was equal to the access time for the main memory unit. As a result, the circuits were very simple, and the simulator functioned well for all the programs.

1. 序 論

本 研究는 simulator라 부르는 특수한 小型 電子計算機를 TTL IC로 設計하여 8 bits one bus line에 連結함으로써 보다 簡便한 指令語(instruction)를 構成하는 方法을 研究하며, 여기에 따른 compiler의 設計方法을 고찰하려 한다.

이러한 compiler를 遂行하기 위하여 사용되는 system clock은 그림 1과 같이 160 nsec 동안 情報가 移動되기 위한 bus line pulse와 실제 micro-operation을 行하기 위한 Cp clock으로 構成되며, 이들 Cp

clock의 몇 개로써 하나의 指令語를 行하도록 하였다.

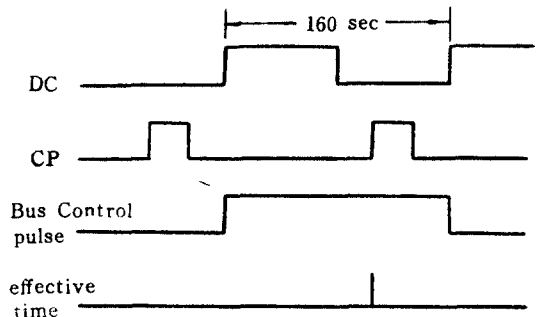


그림 1. 제어 신호
Fig 1. System clock

2. 시뮬레이터의 設計

本 實驗에서 擇한 基本 電子計算機의 構造는 入出力

*,** 正會員, 亞州工科大学 電子工學科
Institut Supérieur de Technologie Franco-coreen "AJOU"
接受日子: 1977年 5月 30日

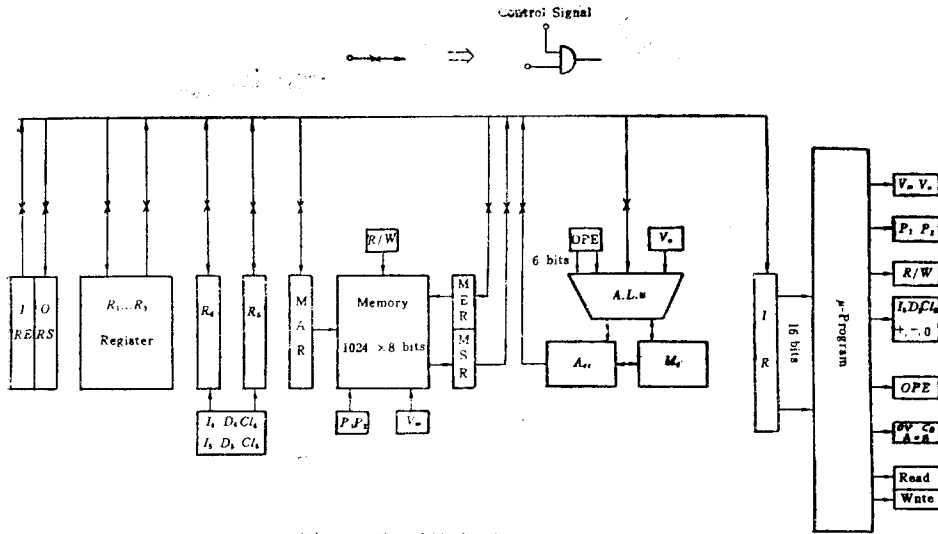


그림 2. 全系統의 계통도
Fig 2. A block diagram for the system

장치, 몇개의 레지스터, 주記憶素子, 演算裝置, 및 micro-program을 制御하는 素子로서 one bus line 방법으로 그림 2와 같이 連結 設計하며 이들을 特性별로 說明하면 다음과 같다.

1) 主記憶素子(main memory unit)

8-bits로 1-word를 構成하며 static RAM memory인 “MOS 2102” 8개를 사용하여 1024 words를 記憶할 수 있으며 256 words의 4區域으로 구분하였고, 이의 memory cycle time은 1μsec이다.

主記憶素子の 補助 레지스터로는 主記憶素子 주소를 指示하는 MAR(memory address register) 10bits register 1개, 記憶素子の 入力 內容을 갖는 MER (memory input register) 8-bits 레지스터 1개, 및 記憶素子の 出力 內容을 갖는 MSR(memory output register) 8-bits 레지스터 1개로써 그림 3과 같이 構成된다.

여기서 MAR 레지스터의 MSB 2bits는 制御裝置로부터 직접 記憶素子の 區域이 선택되며, 基本 access time은 160nsec이므로 memory cycle은 6개의 기본 access time으로 그림 4와 같이 構成된다.

또한 記憶素子를 制御하는 信號는 다음과 같다.

R/W, 1bit; memory에 read, 또는 write하는 信號.

Vm, 1bit; R/W를 有效시키는 信號로서 memory cycle의 初期에 出現.

W, 4bits; bus에 포함된 주소를 指定된 MAR과 MER에 load하도록 制御. 즉, MAR의 주소는 0010,

MER의 주소는 0011이다.

R, 4bits; MSR의 內容을 bus로 傳達하는데 필요한 制御로서 MSR의 주소는 0001이다.

2) 레지스터(registers)

主記憶素子の 기능과 똑같은 역할을 하며 bus line에서 직접 access되고, 8bits로서 構成되며 R1~R5의 5개 레지스터로 W 또는 R 制御信號에 의하여 特定 레지스터의 內容을 bus line에, 또는 bus line으로부터 傳達할 수 있도록 하였으며, 이의 基本 傳達週期는 그림 1에서와 같다.

특히 R4, R5 레지스터는 I4, I5, D4, D5, cl4, cl5 신호에 의하여 1씩 增加, 減少, 또는 clear할 수 있으므로 R4 레지스터를 index register, 또 R5를 program counter register라 부르며, 그림 5에 이들의 構成回路를 보여준다.

또한 부록 II에는 각 레지스터의 R,W 制御 信號에 따른 주소를 보여 주었다.

3) 入出力 裝置(I/O unit)

앞의 다른 레지스터와 같은 역할을 하는 入力 레지스터 RE와, 出力 레지스터 RS가 panel 위에 있는 switches와 lamps에 連結함으로써 二進情報를 취급하도록 하였다.

4) 演算裝置(ALU: arithmetic logic unit)

부록 I에 보여준 48개의 演算動作을 행할 수 있는

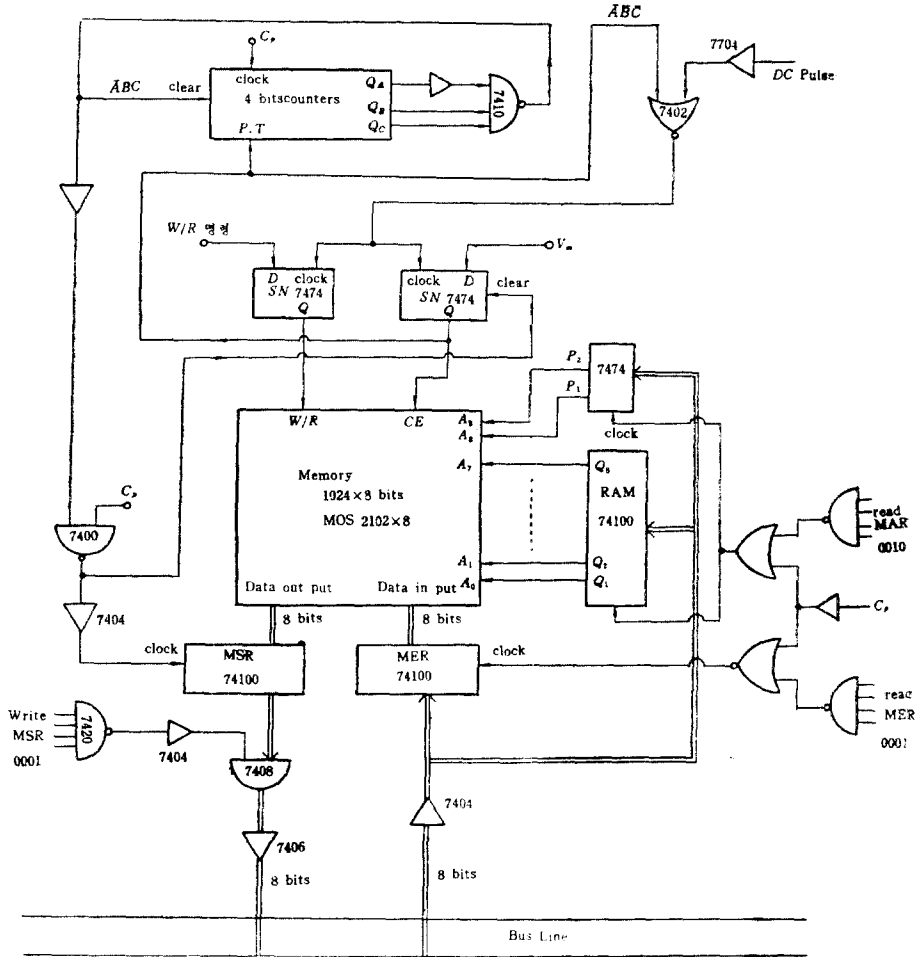


그림 3. 記憶裝置의 構成回路

Fig 3. A circuit for the memory unit

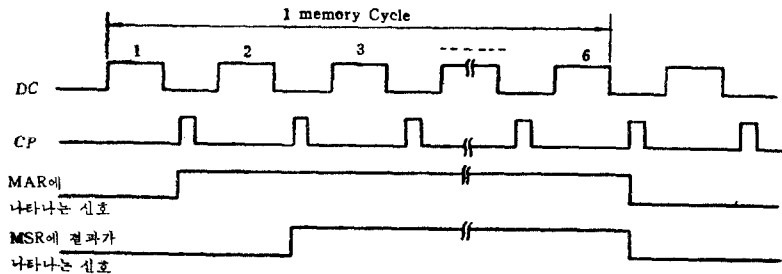


그림 4. Memory cycle time의 timing flowchart

Fig 4. A timing flowchart of the memory cycle time

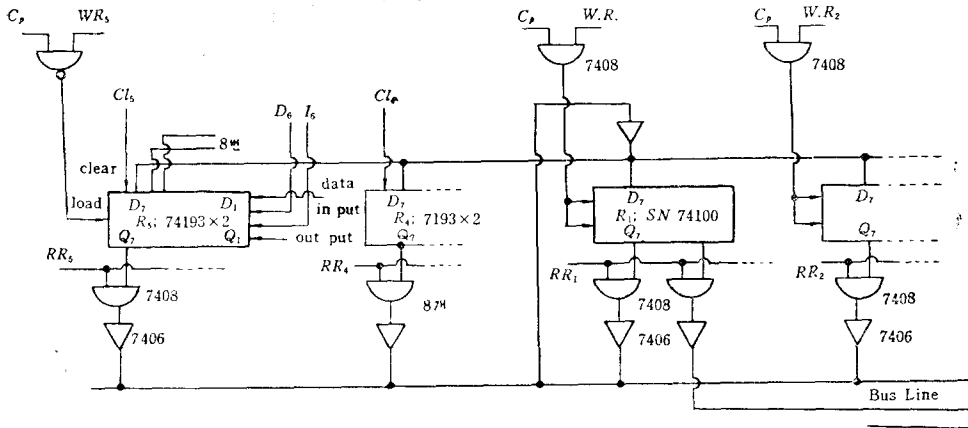


그림 5. 버스에 연결된 레지스터 회로
Fig 5. The register circuit connected on the bus line.

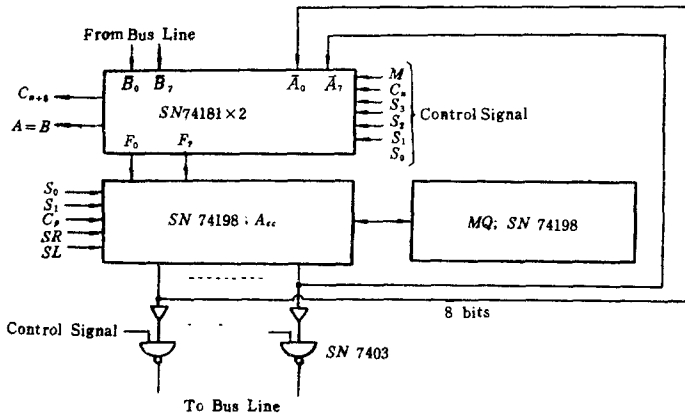


그림 6. SN 74181을 사용한 演算回路
Fig 6. The arithmetic logic unit with the SN74181

SN 74181 2개를 연결하여 그림 6과 같이 구성하였으며 레지스터($R_1 \sim R_5$, I/O registers, MSR) 내용 또는 지정된 主記憶素子 주소의 내용과 Acc(accumulator)의 내용을 演算하여 Acc 또는 MQ 레지스터에 결과를 기억하도록 하였다.

여기에 사용된 Acc와 MQ 레지스터는 SN 74198 shift register를 사용하였다. 演算時間은 加減算에는 1개의 基本週期, 乘除算에는 6개의 基本週期가 필요하며 이때의 制御信號는 다음과 같다.

Vu, 1 bit; 動作週期의 초기에 나타나서 指令語에 주어진 演算動作을 행하도록 한다.

Cov, 1 bit; overflow를 clear 하는 制御信號.

또한 Acc의 상태를 나타내기 위하여 다음 4개의 flip-flop을 사용한다.

OV; overflow를 表示

C; carry를 表示

E; 두 operand가 같음을 表示 즉 $A=B$

Z; Acc의 내용이 0임을 表示.

3. 指令語의 設計

本實驗에서의 모든 指令語는 1 word 16 bits로 構成되도록 하였으며 構成形態는 다음과 같다.

1) Memory reference instruction

OP	R	X	D
0 1	2 5	6 7	8 15

LDA(load)와 STA(store) 두 指令語가 있으며 위 形態로 構成된다.

LDA; operation code는 "01"으로 X와 D로 表示된 memory의 내용을 R에 表示된 주소의 레지스터로

옮기는 언어

STA; operation code는 "10"이며 R 레지스터에 있는 정보를 X와 D로 표시된 memory로 옮기는 언어이며 이때 X가

"00"이면 memory의 첫 구역부터,

"01"이면 R₄에 기억된 주소,

"10"이면 R₅에 기억된 주소,

"11"이면 현재 사용중인 memory 區域內的 D에 표시된 주소이며 D는 변위(displacement)를 표시한다.

2) Arithmetic logic instruction

OP	R	CM	AL code	X
0 1	2	5	6 7 8	13 14 15

op code는 "11"이며 이는 6bits로 구성된 AL code (演算動作)에 따라 加減乘除 및 論理動作을 行하도록 지시하며 M이 0이면 R레지스터의 內容과 Acc의 內容과 演算하고, M이 1인 경우 memory에서 R레지스터에 있는 內容의 주소를 찾아 이의 內容과 Acc의 內容과 演算하게 된다. C가 0이면 Acc의 carry를 무시하고, C가 1인 경우에는 演算에 고려한다. 또한 X의 內容이

"00"이면 進行중인 프로그램을 靜止시키며

"01"이면 R₄내용을 1增加하여 演算(I₄),

"10"이면 R₄내용을 1減小하여 演算(D₄),

"11"이면 R₄내용을 clear한후 演算(c1₄)한다.

3) Input-Output instruction.

OP	R	A	R'	T code
0 1	2	5	6 7 8	11 12 15

op code는 "00"이며 情報을 I/O에 옮기는 言語로서 T code가 "0000"이면 R의 內容을 R'로 옮기고, "0001"이면 R'의 內容을 R로 變換하는 言語이다. 여기서 A가 "00"이면 R'는 入力 레지스터(RE)를 의미하며 "01"이면 R'는 出力 레지스터를 表示한다.

4) Branch instruction

OP	R	A	D
0 1	2	5	6 7 8

op code는 "00"이며 A가 10인 경우 R레지스터 內容이 D가 指示하는 memory의 주소로 jump하고, A가 11이며 D가 指示하는 memory의 內容이 R레지스터로 jump하는 言語를 뜻한다.

4. 콤파일러의 설계

위의 指令語를 行하기 위한 micro-program은 크게 다음의 3狀態로 區分 設計하였다. 즉 memory로부터 指令語를 읽어 IR(instruction register)에 記憶시키는 入力段階(00; fetch phase)와 演算에 필요한 準備作業을 行하는 準備段階(01; preparation phase)와 指令

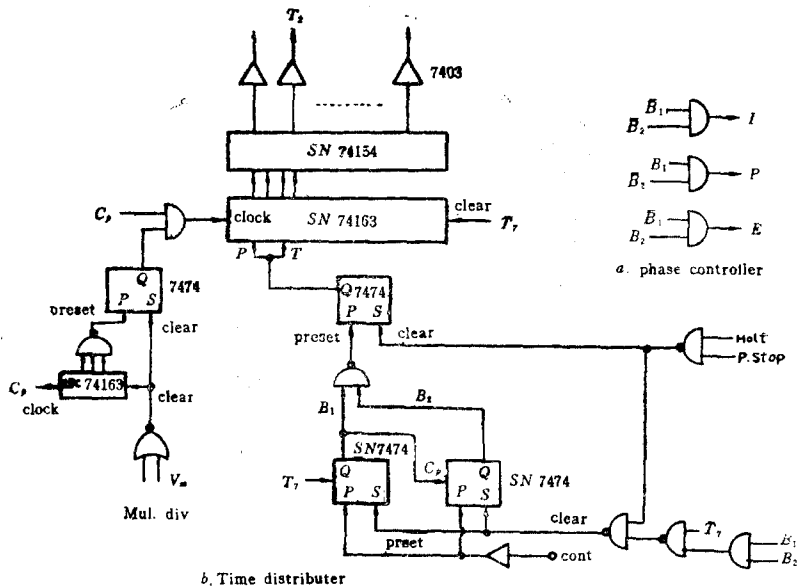


그림 7. 環狀計數回路에 의한 time distributor
Fig 7. The time distributor by the ring counter.

〈표 1〉 a) 입력단계 (fetch phase)

Time	μ -command	comment
I ₁	R→R ₅ , W→RMAR	[R ₅]→MAR
I ₂	At→I ₅ , At→Vm At→[R/W=0]	[R ₅]+1→R ₅ [M]→MSR
I ₃	R→MSR W→IR	[MSR]→IR ₁
I ₄	R→RS W→MAR	[R ⁹]→MAR
I ₅	At→I ₅ , At→[R/W=0] At→Vm	[M]→MAR
I ₆	R→MSR W→IR ₂	[MSR]→IR ₂
I ₇	[0]→I [1]→PorE	준비 또는 수행단계로 변화

b) 준비단계 (preparation phase)

	R ₆ R ₇ =00	R ₆ R ₇ =01	R ₆ R ₇ =10	R ₆ R ₇ =11	comment
P ₁	01001→ AL code At→Vu	010011→ AL code Ak=Vu	010011→ AL code At=Vu	×	[0]=Acc
P ₂	[0,0] →P ₁ P ₂	R→R ₄ 001001→ AL code Ak=Vu	R→RS 001001→ AL code At→Vu	×	주기억소자의 구역선정
P ₂	R→IR ₂ 001001→ AL code At→Vu	R→IR ₂ 001001→ AL code At→Vu	R→IR ₂ 00100→ AL code At→Vu	×	[IR ₂]+ [Acc]→ Acc
P ₄	if Cn+ q=1, At →P ₁ P ₂	if Cn+ q=1, At →P ₁ P ₂	if Cn+ q=1, At →P ₁ P ₂	×	캐리가발생 하면구역변 경
P ₅	0→Cn +b	0→Cn +q	0→Cn +q	×	Cn+q을 Clean
P ₆	×	×	×	×	No action
P ₇	0→P 1→E	0→P 1→E	0→P 1→E	0→P 1→E	수행단계로 변환

〈단 여기서 At는 bus line pulse가 출현함을 의미하고, IR₁, IR₂는 instruction register의 앞부분 8bits와 뒷쪽의 8bits를 구분한 것이며

Time I는 fetch phase a 그림 7의 Time distributor a AND로 연결된 시간을 표시한다.〉

〈단 여기서 준비단계는 operation code가 01, 10인 경우에만 적용되며 주기억소자가 구역을 정하는데 사용된다.〉

c) 수행단계 (execution phase)

	operation code : 01	operation code : 10	operation code : 11				operation code : 00	
			M=0	M=2	T=0000	T=0001	A=00	A=01
E ₁	R→Acc W→MAR	R→ [Register] W→MAR	×	R→Register W→MAR	if A=00 : R→Reg : A=01 -rster : no osct W→100 : -ion	if A=00 action if A=01 R→1100, W →Regi	no R→Register W→MER	×
E ₂	At→ [R/W=0] At→Vm	At→ [R/W=0] At→Vm	×	At→ [R/W=0] At=Vm	×	×	R→Displace ment W→MAR	R→Displa cement W→MAR
E ₃	R→MSR W→MER	R→MSR W→MER	if X=00, HALT X=01, At→I ₄ X=10, At→D ₄ X=11, At→cl ₄	if X=00, HALT X=01, At→I ₄ X=10, At→D ₄ X=11, At→cl ₄	×	×	At→Vm At→ [R/W=1]	At→Vm At→ [R/W=0]
E ₄	R→Register W→MAR	R→Acc W→MAR	R→Register AL code→ ALU At→Au	R→HSR AL code→ ALU At→Vu	×	×	×	R→MSR W→Regist- er
E ₅	A→ [R/W=1] At→Vm	At→ [R/W=0] At=Vm			×	×	×	×
E ₆	×	×	if c=1 [1]=0V	if c=1 [1]=0V	×	×	×	×
E ₇	0→E 1→P	0→E 1→P	0→E 1→I	0→E 1→I	0→E 1→I	0→E 1→I	0→E 1→I	0→E 1→E

語를 實行시키기 위한 遂行段階(10 ; execution phase) 분하였다.

로 區分하고, 이는 2 bits flip-flop의 出力狀態로서 구 자 段階는 7 개의 基本週期로서 構成되는 環狀計數

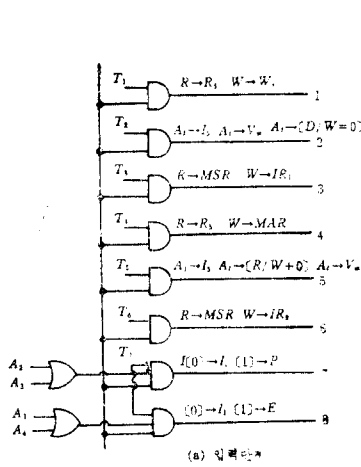
회로(ring counter)에 의하여 動作하며 그림 7과 같이 構成하였다.

여기서 panel 위의 外部制御 端子로는 STOP 단자(進行중인 program을 靜止), CONT 단자(靜止狀態에 있는 program을 다시 動作시킴), HALT 단자(모든 program의 끝을 나타냄)이며 다음 節에서 論하기로 한다.

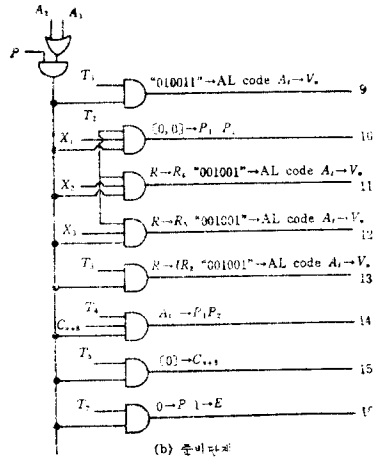
또한 그림 7에서 Vm과 Mul. Div 端子와 連續된 計數器는 乘除算(6基本週期), 또는 記憶端子에서 情報을 읽어 내는데 필요한 時間(6基本週期)을 얻기 위

함이다.

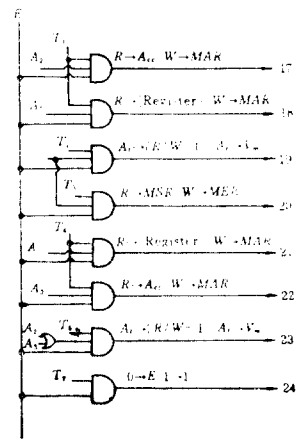
表 1에는 각 指令語를 行하기 위한 micro-operation을 段階별로 구분 說明한 것을 보여준다. 즉 a의 入力段階는 모든 指令語에서 同一하며 記憶端子로부터 8bits의 두 번 읽이 IR(instruction register)에 넣는 過程을 보여주며, b의 準備段階는 10, 01의 operation code에 따른 準備過程을 보여주며 op code가 00, 11에는 이러한 準備段階가 不必要하다. C의 遂行段階는 각 op code에 따라 다르며 이들의 실제 論理回路 構成은 그림 8, 그림 9와 같다.



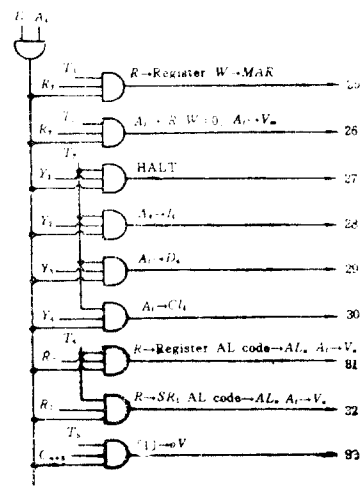
(a) 입력단계



(b) 준비단계



(c) 수행단계



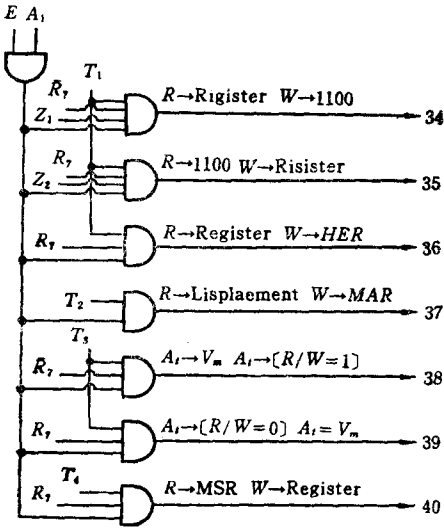
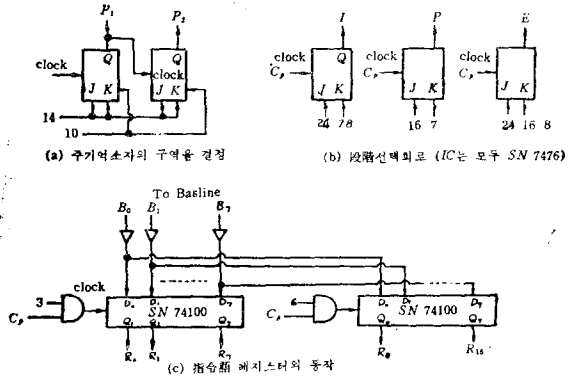


그림 8. 動作回路圖
Fig 8. The operating circuit diagram



(a) 74100 소자의 구역별 연결 (b) 24진수 계수로 (JC는 모두 SN 7476)
(c) 指令語 레지스터의 동작

(c) 指令語 레지스터의 동작

5. 動作 實驗 및 結果

앞면의 panel은 사진 1에서 보듯 16개의 switches와 8개의 붉은 lamps, 8개의 푸른 lamps로 두 그룹으로 8개씩 排列되어 있고, 상 좌측의 4개의 lamps는 C (carry), OV (overflow), Z (Acc의 내용이 0), E (equality, A=B)를 表示한다. 또한 사진 1의 하측 制御 switch의 排列은 그림 10과 같다.

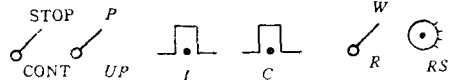


그림 10. 파넬 위의 制御端子
Fig 10. The control terminal on the panel.

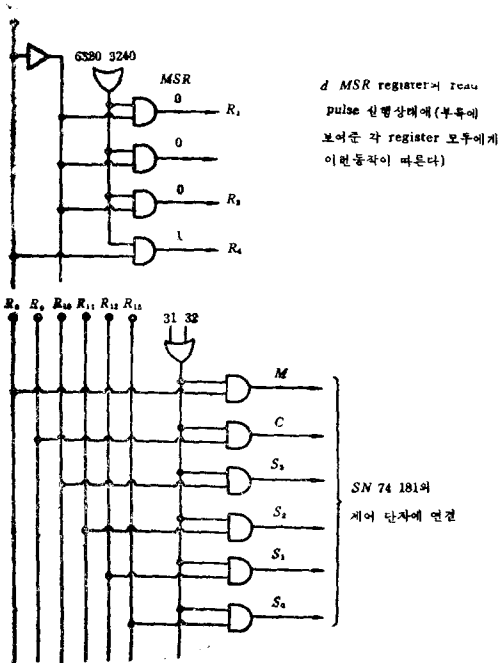


그림 9. 動作連結圖
Fig 9. The connection diagram for the operating circuit

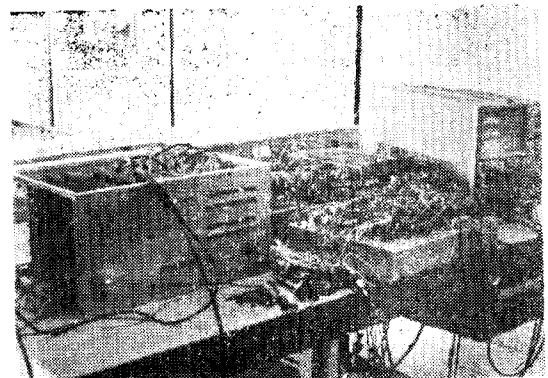
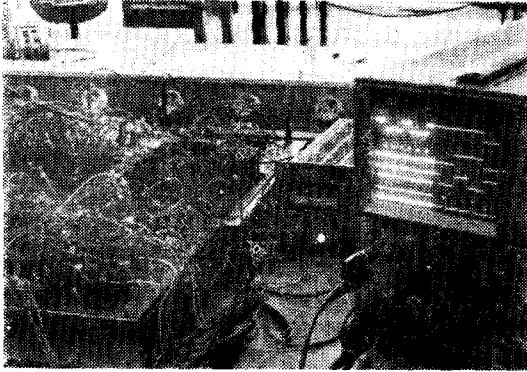
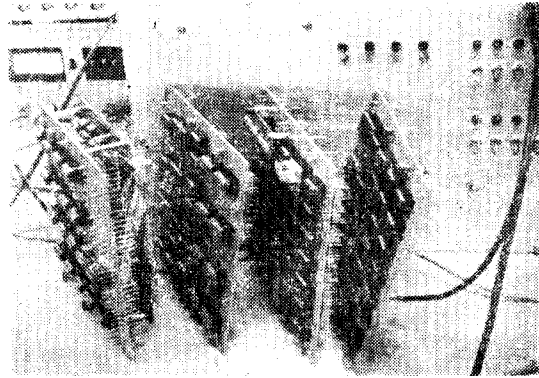


사진 1. 실험 장치 사진
photo 1. The sight for the experiment.



사 진 2. 指令語의 構成回路
photo 2. The real circuit for the micro-instruction.



사 진 3. 시뮬레이터의 内部回路
photo 3. IC boards for the simulator.

그림 10에서 2번째 스위치가 μp 에 있을 때 8개의 푸른 lamps는 主記憶素子의 주소를 나타내고, 8개의 붉은 lamps는 그 주소의 내용을 표시하며 그 밑의 스위치로서 내용을 바꿀 수 있도록 設計하였다. 최전스위치 RS는 이용하고 싶은 主記憶素子의 區域을 결정하며, 두개의 push button인 I와 C는 主記憶素子의 값을 決定하게 한다. 즉 R/W스위치가 W에 있고, C를 누르면 푸른 lamps의 주소로 붉은 lamps의 내용을 主記憶素子에 記錄할 수 있으며, R에 있을 때 C를 누르면 주소가 1의 增加되도록 하였다.

두번째 스위치가 P에 있을 때 앞에서 정의되는 모든 制御信號는 禁止되고 이때 첫번째 스위치의 STOP 위치는 進行中인 program을 中斷하기 위하여 이용되며, CONT 위치는 靜止狀態에 있는 program을 다시

遂行하는 데 사용된다.

두번째 스위치가 μp 에 있을 때 첫번째 스위치는 아무런 의미가 없다.

그러면, 어떻게 program이 遂行되는가를 보여주기 위하여 "00"區域의 100번지와 101번지에 있는 두 수, A=00111000, B=00101010를 더하여 I/O 레지스터에 結果를 넣는 프로그램을 構成해 보자.

먼저 두번째 스위치를 μp , R/W스위치를 W에 놓고 lamps의 스위치와 I, C push button을 사용하여 表 2의 내용을 入力시킨다. 그러면 하면 위의 lamps에는 RS레지스터의 內容(結果)이 揭示된다.

사진 2는 system clock pulse와 指令語의 設計에 따른 實驗裝置를 보여 주며, 사진 3에는 시뮬레이터의 内部 構造를 보여 준다.

〈표 2〉

번 지 수	정 보	주
00000001 00000010	00111000 01100100	00111000을 패널위의 스위치로 주기억소자의 100번지에 load
00000011 00000100	00101010 01100101	00101010을 패널위의 스위치로 주기억소자의 101번지에 load
00000101 00000110	11000001 01001100	Acc의 내용을 Clear
00000111 00001000	01011000 01100100	주기억장치의 100번지에 있는 정보를 R ₁ 레지스터에 load
00001001 00001010	11011001 00100100	R ₁ 레지스터의 내용을 Acc로 load
00001011 00001100	01010100 01100101	101번지에 있는 정보를 R ₂ 레지스터에 load
00001101 00001110	11010101 00100100	[R ₂] + [Acc] → Acc
00001111 00010000	00011101 11000000	[Acc] → RS(Display 하라)
00010001 00010010	11000000 00000000	진행중인 program을 중지(Halt)
01100100 01100101	00111000 00101010	정보값

6. 結 論

小型 電子計算機인 simulator를 設計하는 많은 방법 중, 本 實驗에서는 8bits one bus line system을 擇하였으며, 각 指令語의 길이(field length)를 16bits로 構成함으로써 콤파일러의 設計에는 더욱 간편하게 되었으나, 한편 어분의 비트를 사용하지 않으므로 主記憶素子의 効率的인 이용을 못한 불행한 점도 있었다. 그러나 現在 매우 저렴한 價格으로 供給되는 RAM 보다는 回路 構成의 간단함에 따른 素子(components)의 절약이 더욱 經濟的이라고 생각한다.

또한 SN74181의 演算裝置는 8 자리 加減算에 12nsec <가 필요하지만 本 論文에서는 160nsec의 基本週期를 擇하고 乘除算에는 960nsec가 必要되었다. 물론 이는 참고문헌 7)에 보여준 設計方法에 의하여 演算時間을 단축할 수 있지만 TTL IC의 動作時間에 따른 最適值과 記憶素子의 액세스 타임(access time)을 갈게 하므로 回路를 간단히 하려는 의도였다.

本 論文에서 設計된 simulator는 符號가 없이 절대 값으로만 表示된 8bits(256) 情報에 대하여만 고려되고 one chip으로 된 micro-processor의 性能에 비하

여 뒤떨어 지지만, 그러나 좀더 性能이 改善된 micro-processor를 開發하기 위하여는 이러한 TTL 集積回路에 의한 콤파일러의 構成이 必然的이며 大型 電子計算機의 産業에 매우 중요하므로 많은 研究가 기대된다.

參 考 文 獻

1. Salisbury "Microprogramable Computer Architecture" Elsevier, pp.1~53, 1976.
2. Fredrick J. Hill "Digital systems: Hardware organization and Design" John wiley & sons., pp.15~31, 103~162, 1973.
3. Guy G. Boulaye "Microprogramming" John wiley & sons., pp.34~81, 1971.
4. A. Petitclerc "Traité des ordinateurs" Dunod pp.149~262, 1970.
5. G. Boulaye "Logique et organes des calculateurs numeriques" Dunod, pp.262~318, 1969.
6. "Etude de la constitution d'un calculateur" E, S.E., SE3162, 1974.
7. 박규태, 김용두 "직접회로를 이용한 전자계산기의 연산장치에 관한 연구" 연세논문집 제13집, 1976

부록 I

※ ALU의 arithmetic operation code

SELECTION S ₄ S ₂ S ₁ S ₀	ACTIVE-LOW DATA		
	M=H LOGIC FUNCTIONS	M=L: ARITHMETIC OPERATIONS	
		Cn=0 Cn=O=L	Cn=1 Cn=I=H
L L L L	$F = \bar{A}$	F = A MINUS 1	F = A
L L L H	$F = \bar{A}B$	F = AB MINUS 1	F = AB
L L H L	$F = \bar{A} + B$	F = $\bar{A}B$ MINUS 1	F = $\bar{A}B$
L L H H	F = 1	F = MINUS 1 (2's COMP)	F = ZERO
L H L L	$F = \bar{A} + \bar{B}$	F = A PLUS (A + \bar{B})	F = A PLUS (A + \bar{B}) PLUS 1
L H L H	F = B	F = AB PLUS (A + \bar{B})	F = AB PLUS (A + B) PLUS 1
L H H L	$F = \bar{A} \oplus B$	F = A MINUS B MINUS 1	F = A MINUS B
L H H H	$F = A + \bar{B}$	F = A + \bar{B}	F = (A + \bar{B}) PLUS 1
H L L L	$F = \bar{A}B$	F = A PLUS (A + B)	F = A PLUS (A + B) PLUS 1
H L L H	$F = A \oplus B$	F = A PLUS B	F = A PLUS B PLUS 1
H L H L	F = B	F = $\bar{A}B$ PLUS (A + B)	F = $\bar{A}B$ PLUS (A + B) PLUS 1
H L H H	F = A + B	F = A + B	F = (A + B) PLUS 1
H H L L	F = 0	F = A PLUS A	F = A PLUS A PLUS 1
H H L H	$F = A\bar{B}$	F = AB PLUS A	F = AB PLUS A PLUS 1
H H H L	F = AB	F = $\bar{A}B$ PLUS A	F = $\bar{A}B$ PLUS A PLUS 1
H H H H	F = A	F = A	F = A PLUS 1

부록 II

* 레지스터의 주소.

Unit name	Register name	Address	Read/write control
Memory	MAR	0010	W
	MER	0011	W
	MSR	0001	R
I/O	RE	1100	R
	RS	1100	W
Registers	R ₁	0110	W.R
	R ₂	0101	W.R
	R ₃	0100	W.R
	R ₄	1001	W.R
	R ₅	1000	W.R
ALU.	Acc	0111	R
	MQ	1011	W.R