

# 디지털 시스템과

## 마이크로프로세서설계(I)

金 明 恒\*

### 요 약

이 기술강좌를 통해서 규모가 적지않은 디지털 시스템에 적합한 체계적 설계방법을 설명하며 또 micro-processor 구조와 이것을 응용하는 디지털 시스템설계를 공부한다. 이 첫 강좌에서는 체계적 설계방법의 네 부분인 구조정의, 기계조직, 제어부분, 데이터부분을 소개한다.

이 방법을 응용해서 calculator 설계를 시작한다.

### 1. 체계적 설계방법

디지털시스템은 flip-flop과 gate로 되어있는 순서논리 시스템이다. 순서논리는 state table을 써서 실현화할 수 있다. 그러나 state table를 써서 규모가 큰 디지털 시스템을 설계하는 것은 거의 불가능하지 않으면 매우 힘들 것이다. 그 이유로서는 state의 수가 엄청나게 많기 때문이다. 그래서 규모가 큰 디지털 시스템은 subsystem으로 나누어서 각 subsystem을 register나 counter나 decoder나 multiplexer나 ALU나 control logic 같은 것으로 구성한다. 여러 subsystem의 공통된 data와 control신호를 연결함으로써 디지털 시스템이 형성된다. 컴퓨터의 central processing unit가 하나의 subsystem의 예다.

이러한 subsystem들을 연결하는 방법을 combinational logic이나 sequential circuit 이론으로 표시 못한다. 이러한 이론은 gate와 flip-flop으로 형성된 비교적 간단한 디지털 시스템(MSI로서 shift register 같은 정도)을 설계하는데 쓸 수 있는 것이지 규모가 큰 디지털 시스템을 Boolean function이나 state table로 표시하지 않고 subsystem의 function에 의한 표현을 하기 위하여는 다른 수학적 표시를 써야 한다. 이 목적으로 A Hardware Progr-

aming Language (AHPL) 같은 수학적 표시를 쓸 수 있다. AHPL 표시에서는 register를 디지털 시스템의 기본적 componet로 쓴다. 이러한 표시방법을 씀으로서 register에 있는 data들을 간략하고 정확하게 어떻게 움직이며 또 어떤 processing을 하느냐 하는 것을 표시할 수 있다.

이 설계방법은 다음과 같은 순서로 표시할 수 있다.

1. 구조정의 (Architecture Definition)
2. 기계조직 (Machine Organization)
3. 제어부분 (Control)
4. 데이터부분 (Data Flow)

구조정의에는 두부분이 있다. 첫째는 이 디지털 시스템을 구성하는 register와 다른 소자가 무엇인가를 선언(DECLARE)하는 것이다.

여기서 register라 하면 shift register나 counter나 memory unit 같은 여러가지의 register를 의미한다. Counter는 register로 쓰이는데 또 저장하고 있는 data를 increment나 decrement할 수 있는 역할을 하기도 한다. Memory unit는 많은 register로 형성됐다고 볼 수 있다. Flip-flop한개로 된것은 1-bit register라고 볼 수 있다. 다른 component는 switch나 ALU 같은 것을 포함한다.

구조 정의의 두번째는 디지털 시스템이 어떠한 operation을 하여야 하는가를 표시하는 action을 설명하는 것이다. Action은 디지털 시스템의 operati-

\* 正會員 : 美國 Cornell 大 工 大 電氣工學科 教授 · 工 博

on을 대략 표시하는 것이다. Register에 있는 data는 binary number나 binary-coded decimal (BCD) number나 alphanumeric character나 control information이나 어느 binary-coded information일 수 있다. Action은 register에 있는 data에 어떠한 operation을 한다는 것은 data종류에 따라서 다르다. 숫자는 산술 operation이고 control information은 logic operation이다. Action은 시스템의 operation들이 관계되는 register의 data에 어떻게 된다는 것을 대략 설명하는 것이다.

기계조직 (Machine Organization)은 registers와 다른 소자가 디지털 시스템으로 operation을 하기 위하여 어떻게 연결되어 시스템으로 구성되는 것을 표시한다. 기계조직은 주로 두부분으로 나눌 수 있다. 이것들은 data flow 부분과 control 부분이다. Data flow 부분은 registers 간의 data transfer를 취급한다. 시스템의 operation이 무엇인가에 따라서 data flow 부분의 설계가 결정된다. Transfer는 bus나 다른선을 통해서 시행된다.

Register에 있는 data에 대한 operation을 microoperation이라 한다. Microoperation은 한 clock pulse 동안에 할 수 있는 기본적 operation이다. Operation의 결과는 register에 있는 전의 data를 매치하거나 그렇지 않으면 다른 register에 넣을 수 있다.

Microoperation의 예는 합하거나 빼기, 혹은 clear나, 혹은 clear나, shift나 complement 등이다. 한 microoperation은 이것을 시행하는데 clock pulse가 하나 필요하다. 이러한 microoperation들을 시행하는데 적합한 timing signal를 발생해 내는 부분이 제어부분이다. 디지털 시스템의 모든 operation을 microoperation으로 표시한 것으로 control sequence라 하며 이 control sequence를 시행하기 위한 timing signal를 발생하는 논리회로를 control sequencer라 한다.

컴퓨터는 다목적의 디지털 시스템으로 여러가지 operation을 시행할 수 있으며 여러 operation을 어떠한 순서로 시행하는 것을 instruction으로 되어 있는 program을 써서 지시할 수 있다. 그러므로 컴퓨터를 새로운데 쓸 경우는 단지 instruction으로 되어 있는 새 program을 쓰므로써 할 수 있다. 컴퓨터 설계도 위의 체계적 방법으로 설계할 수 있다.

2. Calculator 설계

이 디지털시스템의 체계적 설계 방법을 calculator 설계를 통해서 자세히 설명하겠다.

이 calculator 설계문제는 다음과 같다.

2.1 구조정의

이 calculator 설계에 있어서 구조정의의를 하여야 한다. 즉 register와 switch등을 아래와 같이 declare한다.

```

DECLARE REGISTER ..... A(4), B(4),
                               IR(2), l(1)
SWITCH ..... SA(4), SB(4),
                               SIR(2), START(1)

```

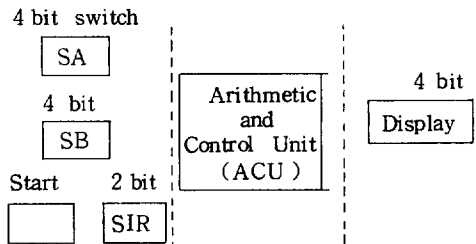
여기서 괄호안의 수는 register의 bit 수를 지시한다. 여기서 l register은 carry로 쓰이며, START는 시작하는데 쓰인다.

이 calculator operation의 action은 다음과 같다.

Operation Code	Action Sequence
SIR=0 $\implies$	$l, A \leftarrow SA + SB$
SIR=1 $\implies$	$l, A \leftarrow SA - SB$
SIR=2 $\implies$	1. $l \leftarrow 0; B \leftarrow SB; ;$ 2. if $B < 0 \neq 0 \Rightarrow$ 끝 또는 if $B < 0 = 0 \Rightarrow B \leftarrow B + 2; ;$ 3. $\rightarrow 1$
SIR=3 $\implies$	$A \leftarrow SA; ;$ $A \leftarrow A \times 2$

Calculator 설계 문제

아래와 같은 세부분으로 되어있는 calculator를 설계하여야 한다.



ACU에 register A와 B가 있으며 switch SA와 SB에서 operand A와 B를 받는다. Register l은 carry를 표시한다.

2 bit switch SIR를 써서 아래와 같은 네 Operation을 표시한다.

OP Code	Operation 정의
0	Operation A와 B를 더해서 register A에 넣는다.
1	Operation B를 operand A에서 빼고 결과를 register A에 넣는다.
2	$\ell$ 를 reset 한다. Operand B를 register B에 넣어라. Register B의 오른쪽 $B_0$ bit이 1이 될 때까지 register B를 right shift 해라.
3	Operand A를 register A에 넣어라. Register A를 2로 곱해라(left shift).

이 때와 같은 TTL Chip을 써서 설계할 것.

- 2 - 74194 4-bit shift registers
- 1 - 74181 ALU
- 1 - 74157 2-to-1 MUX
- 6 - 7474 Dual D flip-flops
- 4 - 7400 2-input NAND

- 2 - 7404 Hex Inverter
- 1 - 7420 4-input NAND
- 1 - 7430 8-input NAND
- 1 - 741 3-input NAND
- 2 - 7408 2-input AND

위의 action은 calculator 설계에 적혀있는 네가지 operation을 설명하는 것이다. 2 bit switch SIR가 0일 때는 이 operation은 4 bit SA와 4 bit SB를 향해서 5 bit의 답을 1 bit  $\ell$  register와 4 bit A register에 넣으라는 것이다. 여기서  $\ell$  register는 답의 most significant bit를 차지하고 A register는 나머지 4 bit를 갖는다. SIR가 1일 때는 SB를 SA에서 빼고 그 결과를  $\ell$ 과 A register에 넣는다. SIR가 2일 때는 첫단계에서(microoperation 1)  $\ell$  register를 0으로 대치한다는 것을 다음과 같이 썼다.

$$\ell \leftarrow 0$$

그리고 동시에 SB switch의 데이터들 B register에 대치한다는 것을 다음과 같이 썼다.

$$B \leftarrow SB$$

여기서 semicolon ;은 이것 양쪽에 있는 microoperation이 동시에 시행된다는 것을 뜻한다. 끝에 semicolon이 두개 ;; 있는데 이것은 이것 양쪽에 있는 microoperation이 동시가 아니고 clock pulse하나 시간차이로 떨어져 있다는 것을 의미한다. microoper-

ation 1과 2가 시간적으로 clock pulse 하나만큼 떨어져 있다는 것이다. microoperation 2에  $B<0>$ 는 B register에 bit 0를 뜻하는 것이다. 즉 B register가  $b_3 b_2 b_1 b_0$ 로 되어 있으면  $b_0$ 를 말하는 것이다. 여기서  $B<0>$ 가 0이면 끝나고  $B<0>$ 가 0이 아니면 B register를 2로 나누어 준다. 이것은 B register를 오른쪽으로 한 bit씩 옮긴다(right shift)는 것이다.

기계조직(machine organization)은 두부분으로 되어 있는데 이것은 제어부분(control section)과 데이터 flow 부분이며 block diagram은 그림 1과 같다. 여기서 register와 switch 사이선이(↓4) 표시는 선이 넷이 있다는 것이다.

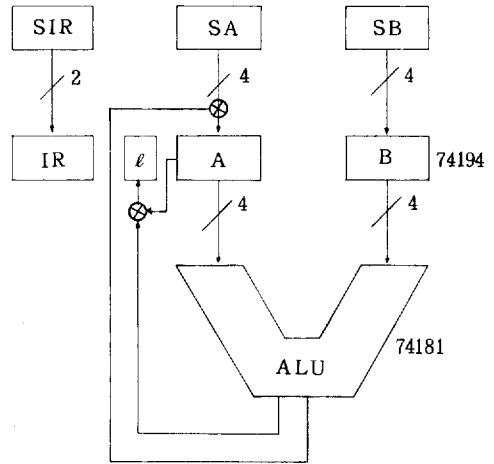


그림 1. Calculator의 block diagram

### 2.2 제어순서(Control Sequence)

이 디지털 시스템의 제어부분 설계를 하는데 항상 operation의 첫단계는 instruction (op code)를 가지고 오는(fetch) 것이다. 그리고 나서 instruction을 decode하고 나서 시행(execute)하는 것이 정상적 순서이다. 이 시스템의 control sequence를 다음과 같이 AHPL로 표시할 수 있다.

1.  $IR \leftarrow SIR ; ;$
2.  $\rightarrow (\overline{IR}_1, IR_1 \wedge \overline{IR}_0, IR_1 \wedge IR_0) / (3, 9, 14) ;$
3.  $A \leftarrow SA ; B \leftarrow SB ; \ell \leftarrow 0 ; ;$
4.  $\rightarrow (\overline{IR}_0, IR_0) / (5, 7) ;$
5.  $\ell, A \leftarrow A + B ; ;$
6.  $\rightarrow 17 ; ;$
7.  $\ell, A \leftarrow A - B ; ;$
8.  $\rightarrow 17 ; ;$
9.  $B \leftarrow SB ; \ell \leftarrow 0 ; ;$

은 microoperation 17에 가라는 것이며 2는 세가지 조건에 따라서 3이나 9나 14에 가라는 것이다. 여기서 instruction register 의 두번째 bit  $IR_1=0$  ( $IR_1=1$ )이면 3으로 가고  $IR_1 \wedge IR_0=1$  (즉  $IR_1=1$  과  $IR_0=0$ )이면 9로 가고  $IR_1 \wedge IR_0=1$  (즉  $IR_1=1$  과  $IR_0=1$ )이면 14로 가라는 것이다. 이 제어순서가 이 디지털시스템의 operation을 정확히 microoperation으로 표시한 것이다.

이 제어순서에 필요한 control sequencer 와 데이터 부분 설계는 다음 강좌 [6]에서 계속할 예정이다.

- 10.  $\rightarrow (B_0 VB_1 VB_2 VB_3, \overline{B_0} \overline{VB_1} \overline{VB_2} \overline{VB_3}) / (11, 17)$
- 11.  $\rightarrow (B_0, \overline{B_0}) / (17, 12) ;$
- 12.  $B, \ell \leftarrow \text{right shift } (B, \ell)$
- 13.  $\rightarrow 11 ; ;$
- 14.  $A \leftarrow SA ; \ell \leftarrow 0 ; ;$
- 15.  $\rightarrow (A_0 VA_1 VA_2 VA_3, \overline{A_0} \overline{VA_1} \overline{VA_2} \overline{VA_3}) / (16, 17)$
- 16.  $\ell, A \leftarrow \text{left shift } (\ell, A)$
- 17. Halt

Microoperation 2, 4, 6, 8, 10, 11, 13, 15은 branch operation이다. 그중 microoperation 2는 condition-

all branch 이고 6은 unconditional branch이다. 6

Reference

- [1] F. Hill and G. Peterson; "DIGITAL SYSTEM: Hardware Organization and Design," 2nd Ed., John Wiley & Sons, Inc., 1978, pp. 121-402.
- [2] M. Mano; "Digital Logic and Computer Design," Prentice-Hall, Inc., Englewood Cliffs, N.J., 1979, Chapter 8.
- [3] Mano, M.M.; "Computer System Architecture," Prentice Hall, Inc., Englewood Cliffs, N.J., 1976.
- [4] Kline, M.M.; "Digital Computer Design," Prentice-Hall, Inc., Englewood Cliffs, N.J., 1977.
- [5] Chu, Y.; "Computer Organization and Microprogramming," Prentice-Hall Inc., Englewood Cliffs, N.J., 1972.
- [6] 김명환, "디지털 시스템과 마이크로 프로세서 설계: II, 체계적설계-2" 전기공학회지 (발행 예정).

會 員 動 靜

成樂正會員은 스위스의 로잔느에서 열린 제 8차 유럽原子力産業會議에 특별초청연사로 참석후 귀국 (6月 19日 ~ 30日)

白英鶴會員은 韓國電氣通信研究所 所長으로 취임

李承院 (前會長) 會員은 超電導研究次日本에 1個月 豫定으로 訪門

鮮于學永 (理事) 會員은 大韓電氣協會 産業視察團 一員으로 美國방문후 귀국

盧 澗 來 會員은 韓電 원자력 발전부에서 월성 원자력 발전소장으로 옮김

盧 泰 天 會員은 忠南大 工業教育大 技術教育科로 옮김