

Microprocessor 에 關한 一般의 考察

余 在 興*

(Yuh, Jae Heung)

1. 序 論

複雜한 科學計算이나 大體한 量의 情報處理 및 多樣한 自動調節制御裝置에 으러히 起用되는 電子計算機는 文字 그대로 現代文明의 寵兒가 되었다. 一般의 大型 電子計算機는 그 設置 및 使用에 있어서 經濟的으로 매우 큰 負擔을 요하기 때문에 比較的 低廉한 小型 電子計算機(Minicomputer)가 開發되었으며, 뒤이어 小型 電子計算機의 採用이 不可缺할 만큼 處理되어질 情報量이 많지도 않을 뿐만 아니라 그 處理速度가 빠를 必要가 없는 일에 適合하며 높은 信賴性, System Interface 와의 融通性 및 Modular 性을 지니고 價格 또한 低廉한 Processor 의 開發이 要求되게 되었다. 따라서, 지난 수년 동안 놀랄 만큼 急速히 發達한 MOS/LSI 技術에 힘입어 몇몇 半導體 集積回路 生産業體들은 몇개의 LSI Chip 을 使用하여 多方面으로 應用價値가 많은 Processor 를 開發하게 되었는데 이것이 바로 Microprocessor 이다. 現在까지 開發된 Microprocessor 의 큰 缺點은 兩極性回路(bipolar circuit)로 構成된 小型 電子計算機에 比하여 命令遂行時間(Instruction Execution Time)이 길다는 것이 첫째이며, 둘째로는 歷史가 짧은 關係로 Software Library, Interfacing, Field Service 等の 完熟한 應用補助가 大體的으로 不充分하다는 點이다. 그러나, Microprocessor 는 Programming Language 를 Microprogramming Technique 에 의하여 任意로 바꿀 수 있고 Modular 形式의 融通性을 갖고 있어서 그다지 많은 Memory 를 要求하지 않는 Process 로 널리 利用되며, 따라서 大型 및 小型電

子計算機의 周邊裝置制御, Batching Process, Calculator, Point-of-sale Terminal, Numerical and Process Control, 計測器, 醫學電子機器, 電子計算機 Terminal, OCR, CRT 및 Video 장치, 電子交換, 交通制御, 信號分析, 綿織物의 色彩分析 等 이외에도 여러 方面에 널리 利用되고 있다.

Microprocessor 는 두 가지 型으로 크게 分類할 수 있다. 첫째로는 8 또는 16비트(bit) 語長의 데이터를 從來의 小型 電子計算機와 같이 並列處理하는 型이며, 또 하나는 4비트 BCD 데이터를 Calculator 와 같이 直列處理하는 型이다.

BCD 直列處理型은 비록 命令遂行時間이 긴 편이지만, 可變語長方式이므로 融通性이 큰 利點이 있다고 하겠다.

Microprocessor 의 記憶裝置로는 주로 半導體 RAM(Random Access Memory)과 ROM(Read Only Memory)이 使用된다.

Microprocessor 의 모든 制御는 大部分 制御 ROM 에 記憶되어 있는 Microprogram Instruction 에 의하여 遂行된다.

2. 實用化된 Microprocessor 의 例

1971年 美國의 Intel社에서 開發된 MCS-4 Micro Computer Set 를 嚆矢로 여러 半導體集積回路生産業體로부터 자기 特色있는 製品들이 生産販賣되고 있는데 그 代表的인 例로 American Micro-system社의 AMI 7200, National社의 MAPS 와 GPC/P, Fairchild社의 PPS--25, Rockwell Microelectronics社의 Model 10660, 그리고 Intel社가 MCS-4의 뒤를 이어 내어 놓은 MCS-8 등이 있다. 이들은 제각기 特有한 長短點을 지니고 있지만 여기서는 AMI 7200과 MCS-4

* 韓國科學技術研究所 方式機器研究室

Korea Institute of Science and Technology

그리고 MCS-8에 對하여 簡單히 살펴 보기로 한다.

2·1 AMI 7200

American Micro-system社에서 開發된 이 모델은 세계의 MOS/LSI Chip으로 構成되어 있는 3番地方式의 Processor이다. 이 모델은 8비트 並列處理機能을 가지고 있으며 16비트 並列處理를 위하여서는 레지스터(register) 및 ALU Chip을 하나 더 追加시키면 된다. 가장 注目할 만한 點은 32개의 레지스터로 構成되어 있는 Push-Down Stack이 있다는 점, Bus System의 應用이 便利하다는 점 및 Microinstruction의 遂行에 pipeline 方式을 採擇하였다는 점이다. LIFO(Last In First Out) 方式으로 運用되는 總 32個의 Register Stack은 Real Time 또는 프로그램 遂行中에 發生하는 Interrupt나 Subroutine 處理를 할 때 Program Counter의 內容과 Status Register의 內容을 待避시키는 場所로 사용되기도 하며 12個의 一般 Register와 같이 一般用途로도 사용될 수 있다.

Bus는 Originating Device의 Bus 사용要求가 있을 때부터 Destination Device에서 情報受信準備信號(ready)가 있어서 情報의 授受가 完了될 때까지 第三의 Device가 Bus를 사용할 수 없게 되는 Handshake 方式을 採擇하고 있으며, 따라서 同一한 Bus에 連結되어 있는 두개 이상의 裝置가 同時에 Bus 사용을 要求할 때에는 優先權順位에 따라 Bus 사용이 決定된다.

Data Exchange Bus는 記憶裝置, Processor, 制御 Pannel 및 入出力裝置 사이에 並列情報을 傳達하는 데 사용된다.

基本動作順序는 記憶裝置로부터 Micro-control Chip으로 命令이 전달되는 것으로부터 順次的으로 遂行된다.

Microinstruction ROM은 24비트語長의 512單語로 構成되어 있으며 하나의 24비트 Microinstruction은 兩分되어 所謂 Pipeline 方式에 의하여 傳送된다. 換言하면, 前段의 12비트는 두 개의 演算 Register를 指定하고 나머지 後段 12비트는 ALU 演算과 演算結果를 貯藏하는 Destination Register를 決定지어 줌으로써 뒤 12비트가 遂行

되는 중에 다음 Microinstruction의 前段 12비트를 重疊的으로 遂行함으로써 Register의 內容을 Access 하는 데 所要되는 比較的 긴 時間을 節約할 수 있다.

2·2 MCS-4와 MCS-8

이 모델들은 Intel社가 開發한 것으로서 MCS-4는 45個의 命令을 使用하여 4비트單位の 데이터를 並列處理하며 演算은 二進 또는 十進으로 行하여지고 있다. 命令遂行時間은 $10.8\mu s$ 정도이어서 두 개의 8디지트 數字의 合算은 $850\mu s$ 에 處理된다. ROM은 $4KW \times 8bit$ 까지, 그리고 RAM은 $1280W \times 4 bit$ 까지 收容할 수 있으며 interface logic을 使用하지 않고도 128個의 I/O Channel을 設置할 수 있다는 큰 長點이 있다. Machine Cycle은 모두 8個의 Clock Cycle로 構成되는데 CPU에서 同期信號인 Sync Pulse를 ROM 또는 RAM에 傳達함으로써 始作된다. Sync Pulse 發生後 ROM(또는 RAM)은 12비트의 ROM(RAM)番地를 750 KHz Clock의 세 Cycle 동안에 Data Bus로부터 받아들이며 그 동안에 Program Counter의 內容은 하나 增加된다. 이어서 다음 두 Clock Cycle 동안에는 選定된 ROM(또는 RAM)番地에서 얻어지는 8비트의 命令 또는 데이터가 Instruction Register의 OPR(operation field)과 OPA(address field)에 보내지며, 나머지 세 Cycle 동안에 命令이 遂行된다.

MCS-8은 MCS-4의 뒤를 이어 開發된 8비트 並列處理型 Processor인데, 어떤 型의 半導體記憶素子라도 最大 $16kW \times 8bit$ 까지 사용할 수 있기 때문에 記憶容量이 MCS-4보다 훨씬 크다. MCS-8 microprocessor에 使用된 CPU(8008)에는 6個의 8 Bit Data Register와 하나의 8 Bit Accumulator, 8비트의 臨時레지스터 2個, 4個의 Flag Bit 그리고 8비트 並列演算部가 있다. 14비트의 Program Counter와 14비트 語로 되어 있는 7W의 Memory Stack은 프로그램과 Subroutine番地를 保管하는데 使用된다.

Processor內的 모든 데이터는 8個의 데이터 또는 番地비스($D_0 \sim D_7$)를 통하여 非同期的으로 記憶裝置와 交流된다.

前述한 바와 같이 CPU內的 7Word Memory

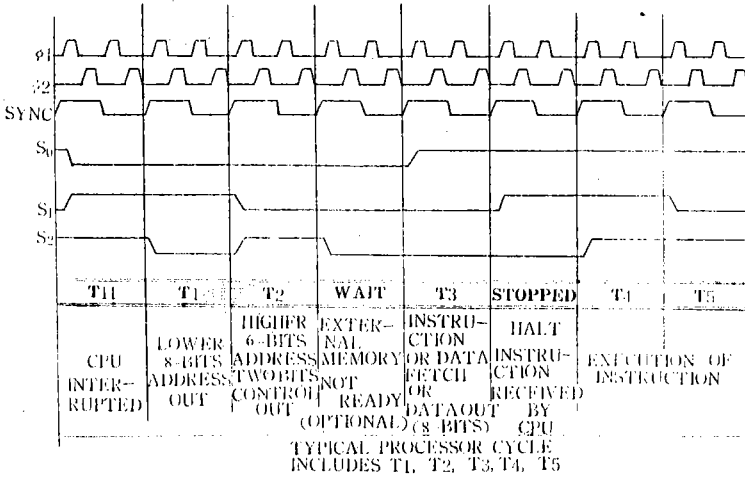


그림 1. 8008 Instruction Cycle

Stack 은 Subroutine 의 Return Address 保管用으로 쓰이기 때문에 最大 7회까지 Subroutine Nesting 이 可能하게 된다. Scratch Pad Memory 에는 Accumulator 와 8비트 레지스터 6個가 있는데 이 중에서 두 개의 레지스터가 합하여 하나의 Index Register 로 使用된다.

Memory Addressing 은 14비트로 하기 때문에 全體 16k 語의 記憶裝置는 直接番地(Direct Addressing) 方式으로 呼出(access)될 수 있다. Clock 周波數는 最大 500KHz 로 擇할 수 있어서 Non-memory Reference 命令은 約 20 μs 內에 處理되어 진다.

CPU 는 Interrupt 制御線을 使用하여 Interrupt 를 할 수 있기 때문에 速度가 낮은 I/O 周邊裝置에 對한 서어비스는 主 프로그램(main program) 遂行中에도 行할 수 있게 된다.

Ready 信號는 種類나 速度에 關係 없이 어떠한 半導體記憶素子라도 使用할 수 있도록 하기 위하여 CPU 와 Memory Cycle 의 同期를 맞추기 위한 信號이다.

그림 1은 8008의 Instruction Cycle 을 圖示한 것이다. 보통 Machine Cycle 은 다섯 개의 狀態로 分離된다. T1 과 T2 동안에는 Memory Address 가 記憶裝置로 보내지며, T3 때 記憶裝置로부터 命令이나 데이터를 가져와서 T4와 T5 동안에 命令을 遂行한다. 記憶裝置가 데이터를 받거나

提供할 準備狀態가 되어 있지 않을 때 Processor 는 WAIT 狀態에 들어간다.

MCS-8에서 使用되는 命令은 세 가지 形式으로 되어 있는데, 1바이트命令形式(1-byte instruction format)은 Register 와 Register 사이의 動作, Memory Reference, 入出力演算, Rotate 또는 Return Instruction 等に 使用된다. 그리고, 2바이트命令形式은 Immediate Mode 에 使用되며, 3바이트命令形式은 Jump 또는 Subroutine Call 命令에 使用된다.

MCS-8이 MCS-4 보다 좋은 點은 語長이 길기

	MCS-4	MCS-8
데이터 處理方法	4 bit 並列 十進 또는 二進	8 bit 並列 二進
Subroutine Nesting	3회까지 可能	7회까지 可能
外部裝置에 對한 Service 方法	外部信號를 테스트	Interrupt 處理
記憶裝置運轉	同期式	非同期式
記憶裝置와 入出力裝置를 爲한 Interface	不必要	約20個정도의 TTL/IC 必要
記憶容量	4K×8 bit ROM 1280×4 bit RAM	16K×8 bit
Minimum System	CPU, RAM	CPU, ROM, 20 TTL/IC

때문에 많은 량의 데이터處理에 便利하며 MCS-4에 없는 여러 가지 命令을 遂行할 수 있는 點이다. 특히 Real-time Operation 이나 Data Communication에 適合한 Priority Interrupt가 可能하다는 點은 큰 長點이라 하겠다.

MCS-4와 MCS-8을 비교한 표는 앞의 표와 같다.

3. Microprocessor의 設計

Microprocessor System 設定에 있어서 고려되어야 할 가장 重要한 問題는 現在 標準化된 小型 電子計算機와의 競爭에서 뒤떨어지지 않기 위하여, 첫째 價格이 低廉하고, 둘째 命令遂行時間을 可能한 限 짧게 하여야 한다는 點이다.

이를 위하여서는 먼저 從來 使用되던 逐次的 Random Logic 대신에 Microprogramming 方式을 採擇하여 命令을 遂行하도록 設計할 필요가 있다. 勿論, Microprogramming 方式은 Random Logic 方式보다 Machine Instruction 遂行時間이 길어

지는 短點이 있지만 命令의 種類를 增加 또는 減少시킬 必要가 있을 때 Microprogram의 內容과 크기를 變更함으로써 쉽게 解決될 수 있을 뿐만 아니라 소위 Emulation이 容易하다는 큰 長點과 무엇보다도 Processor의 價格을 低廉하게 할 수 있다는 가장 큰 利點이 있기 때문이다 (그림 2 참조). 둘째로, MOS/LSI 外에 必要한 兩極性 IC(例: TTL)의 數를 可能한 限 줄이는 것이 必要하며, 셋째 Software 개발을 最小限으로 줄이기 위하여 TTL로 計設된 既存 小型電子計算機의 命令세트(instruction set)를 引用할 必要가 있다. 끝으로 시스템의 融通性을 考慮하여 多方面에 應用하기 容易하도록 設計되어야 하겠다.

實在으로 시스템을 몇개의 Chip으로 區分할 것인가를 決定하기 위하여서는 먼저 IC 包裝에 있어서 Pin 數의 制限을 考慮하여야 하며, 또한 어떻게 區分하는 것이 가장 効果的인가 하는 問題를 念頭에 두어야 한다. 가령 Register와 演算部를 獨立시키는 경우 發生하는 Chip과 Chip 사이의 傳達遲延(transition delay) 問題를 Pipeline 方式으로 解決할 것인가 또는 비록 費用이 더드는 40Pin 包裝을 使用하여서라도 Register와 演算部를 統合할 것인가를 決定하여야 한다.

一般目的用 Microprocessor System을 設計하기 위하여 考慮되어야 할 여러 가지 問題點이 있겠지만 여기서는 Microprogram에 關한 大體的인 檢討를 하여 보기로 한다.

3.1 Microprogramming

論理回路設計問題를 Microprogramming 方式으로 解決하기 위하여 우선 設計方向을 Micro-concept에 맞도록 設定하여야 한다는 點이 무엇보다도 重要하다. Micro-concept는 任意의 한 Instruction(命令)을 遂行함에 있어 聯關된 모든 論理回路들간의 調節制御 및 逐次化(sequencing)를 또 하나의 다른 Program Sequence로 假想하여 모든 命令들을 Programming으로 遂行하는 概念이다. Microprocessor의 Microprogramming은 임의의 두 Register 內容을 演算하여 그 結果를 어떤 Register에 넣는다든가 또는 Register 사이의 演算에서 얻어진 結果를 檢査하여 次期의

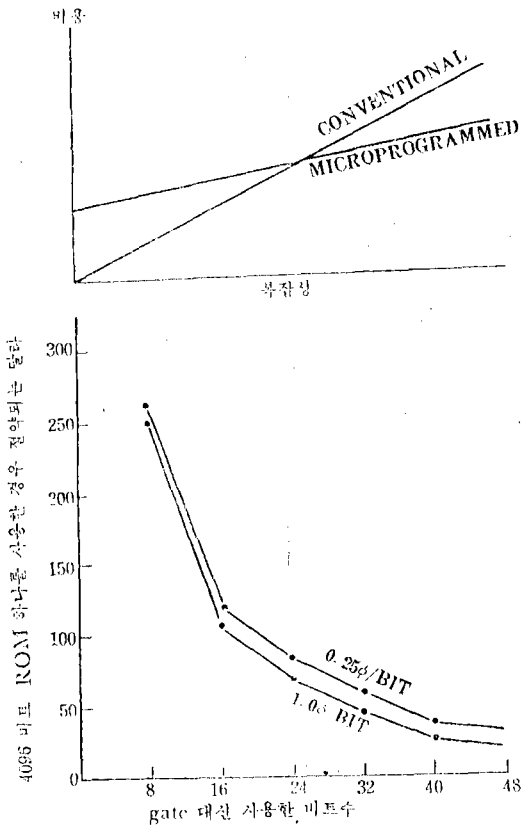


그림 2.

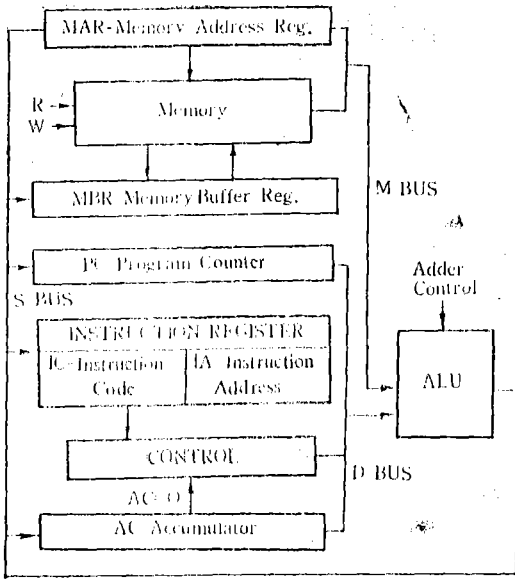


그림 3.

動作을 選定하는 것과 같이 Register 의 內容을 處理하는 데 適合하도록 프로그램되는 것이 大部分이다. 따라서, 모든 論理回路는 이러한 一連의 動作에 맞도록 꾸며야 한다.

Microprogram 裝置의 cycle time 은 그 機械의 命令遂行時間보다 5倍정도 빠르게 設計하는 것이 普通이다.

Microprogram 은 一般의 呼出時間(access time)이 빠른 ROM 內에 넣어 놓는데, Microprogramming 을 하기 위하여 慎重히 考慮할 點은 하나의 Microinstruction 을 몇개의 비트로 構成할 것이며 그 命令形式을 어떻게 設定할 것인가하는 問題이다. Microprocessor 에 使用되는 Microinstruction 은 普通 16비트 또는 32비트로 構成되어 있다. 그림 3에 簡單한 Microprogram 方式의 機械에 關한 構成圖와 Microprogram 을 使用하여 記憶裝置의 內容을 Accumulator 에 合算

표 1.

		Memory Control	JMP Instr.	Adder Control	M Bus Control	D Bus Control	S Bus Control	JMP Address Field
Instruction Fetch	1			DTS		PCD	SMAR	
	2	R						
	3			MTS			SIR	
Instruction Decode	4		JOPC					
Execute	5			DTS		IAD	SMAR	
	6	R						
	7			PLUS	MBRM	ACD	SAC	
Return	8			ONE		PCD	SPC	
	9		JMP					START

- R: 기억장치 內容을 MBR 로 보냄
- DTS: D BUS 에 실린 데이터를 가산기를 통하여 S BUS 에 실림
- PCD: PC 의 內容을 D BUS 에 실림
- SMAR: S BUS 의 內容을 MAR 로 보냄
- MTS: M BUS 에 실린 데이터를 가산기를 통하여 S BUS 에 실림
- MBRM: MBR 의 內容을 M BUS 에 실림
- SIR: S BUS 의 內容을 IR 에 보냄
- JOPC: OP CODE FIELD 의 번지로 JUMP
- IAD: IA 의 內容을 D BUS 에 실림
- PLUS: M BUS 內容과 D BUS 內容을 합산하여 S BUS 에 실림
- ACD: AC 의 內容을 D BUS 에 실림
- SAC: S BUS 의 內容을 AC 로 보냄
- ONE: D BUS 의 內容을 가산기로 통하여 하나 증가시켜 S BUS 에 실림
- SPC: S BUS 의 內容을 PC 로 보냄
- JMP: Micro Instruction 의 번지부에 있는 번지로 립

하는 program step 을 表示한 것이 표 1이다.

3.2 Microprocessor 의 Timing

Microprocessor 의 設計上 考慮되어야 할 다른 하나의 重要한 問題는 Timing Scheme 을 어떻게 策定하는가이다. 일정한 시간 동안 處理될 수 있는 命令의 量은 Microinstruction 의 Control Cycle 이 얼마나 빠른가에 의하여 比例的으로 增加하여 Control Cycle 이 充分히 빠른 條件에서는 特定한 機能을 遂行하는 制御비트의 數를 줄일 수 있고 따라서 Microinstruction 語와 Subroutine 들의 效率이 向上될 뿐만 아니라 Microprogram 自體內의 Loop 를 使用하는 頻度를 크게 할 수 있게 되는 利點이 생긴다. 다시 말하면 Microinstruction 의 語長을 줄이고 또한 Subroutine 및 Loop 를 사용하여 必要한 命令數(microinstruction step)를 縮小시키면 相對的으로 Macroinstruction 의 處理時間이 늦어지는데 이것을 補償하기 위하여서는 Control Cycle 을 빠르게 하여야 한다는 問題가 있지만 이렇게 함으로써 制御 ROM 의 크기를 줄일 수 있게 되는 利點이 생긴다. 따라서 Microprogram 을 할 때에는 命令處理速度와 ROM 의 크기를 결정하는 데 慎重을 기하여야 한다. 그림 4 는 Microcontroller 의 Timing 關係를 表示한 一例이다. 普通 Microinstruction 의 遂行은 먼저 ROM 의 特定番地內容이 Strobe 되면 ROM Program Counter 가 하나 增加되고 Strobe 된 內容에 該當되는 Micro-operation 이 遂行되고 그 結果를 檢査하여 條件附 壼(conditional jump)에 對한 決定을 한 後에 다시 다음 ROM 番地의 內容을 呼出하는 順序를 밟는다.

Control Cycle 을 빠르게 하기 위한 하나의 方法으로 現在 呼出된 命令을 遂行하는 中에 다음

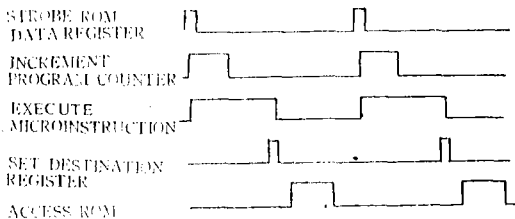


그림 4.

Microinstruction 을 呼出하는 所謂 Overlapping 方法이 있는데 이 경우의 Timing 은 그림 4와는 매우 다르게 된다.

ROM 의 크기를 줄이기 위하여 Control Cycle 을 빠르게 하는 方法外에 從來에 使用되던 復號化方式(decoding method)도 混用될 수 있겠으나 그렇게 되면 Microprogram 을 使用하는 根本的인 目的에 벗어나므로 그 限界를 定하여 두는 것이 좋겠다.

4. Microprocessor 의 經濟性 및 將來

近來의 大型電子計算機 設計는 CPU 를 多重化하여 所謂 Parallelism 과 Pipeline 方式을 採擇하는 動向이 있다.

그 卑近한 例로 Illinois Institute of Tech. 에서 開發한 Illiac 4는 64個의 Processor 를 가지고 있어서 각 Processor 들은 제각기 別도의 記憶裝置와 制御裝置를 가지고 있다. 또 Control Data 社의 Star-100은 여러 개의 Processor 가 多量의 業務를 分割하여 한 개의 Processor 의 處理出力을 다른 하나의 入力에 加하여 連續的으로 處理하는 Pipe Line 方式을 採擇하고 있다. 위와 같이 여러 개의 Processor 를 使用하면 그의 費用도 따라서 急增하게 되므로, 近來에는 小型電子計算機의 應用에 對한 檢討가 活潑히 進行되고 있다.

이와 때를 같이 하여 MOS/LSI 技術도 急進的으로 發展하고 있어서 半導體 MOS/LSI 製造業體는 Processor Cycle Time 이 더 빠르고 價格이 低廉한 Microprocessor 를 開發하는 同時에 이에 對한 Software 開發도 活潑하게 展開하고 있어서 Microprocessor 의 將來는 매우 밝다고 하겠다.

從來의 Microprocessor 는 大部分 p-Channel MOS/LSI 로 構成되어 있지만 MOS/LSI 製造技術이 發達됨에 따라 n-Channel MOS/LSI microprocessor 의 開發이 推進中이며, CMOS(Complementary MOS)技術이 점차 發展되면 現在의 Dynamic Logic 構成에서 Static Logic 構成으로 바뀔 뿐만 아니라 Processor 의 速度 또한 한층 빠르게 될 展望이 보인다.

最近報道에 의하면 1975년까지 數十萬키트의 Microprocessor 가 販賣될 것으로 豫想된다고 하

며 그 價格은 大量購入時에 세트當 \$30정도로 될 추세이며 組立된 Prototype은 開發된 Software와 함께 \$400에서 \$2,000 정도에 販賣될 것이라 한다. 따라서 價格面으로 보면 小型電子計算機와 比較가 안되리만큼 低廉한 것이다.

아직까지는 Microprocessor가 小型電子計算機에 比하여 速度나 融通性에 있어서 많이 뒤떨어져 있지만, 앞으로 小型電子計算機의 Processor나 周邊裝置制御용으로 크게 脚光을 받을 것임에 틀림이 없다.

Microprocessor의 世界市場을 席卷하고 있는 Intel社는 1973年 후반에 n-channel MOS/LSI를 使用하여 8비트 單一 Chip CPU를 開發할 豫定이라 하는데 命令遂行時間은 約 $2\mu\text{s}$ 정도가 될 것이라 한다. 이것은 從來 p-Channel Chip 보다 10倍 이상 빠르며 小型電子計算機의 $1/3\sim 1/2$ 의 速度이다.

이러한 추세로 미루어보아 Microprocessor의 應用範圍는 더욱 넓어질 것이 明白하다.

참 고 문 헌

- 1) Gerald Lapidus: "MOS/LSI Launches the Low-cost Processor" *IEEE Spectrum*(Nov. 1972)
- 2) William H. Davidow: "General-Purpose Microcontrollers" *Computer Design*(July/Aug. 1972)
- 3) Wallace B. Riley: "Wanted for the 70's Easier-to-program Computers" *Electronics* p. 61~p. 84 (Sep. 13, 1971)
- 4) George Sideris: "Microcomputers Muscle in" *Electronics* (March 1, 1973)
- 5) Jack L. Rosenfeld: "Micromultiprocessing: An Approach to Multiprocessing at the level of Very Small Tasks" *IEEE Transaction on Computers*, C-22(2), p. 149~p. 153 (Feb. 1973)
- 6) Intel Corporation: MCS-8 Technical Manual (1972)