

技術解説

마이크로 프로세서

朴 喆 熙\*

디지털 컴퓨터의 중앙처리장치(CPU)와 비슷한 기능을 수행하는 범용적 고집적회로(LSI)인 마이크로-프로세서( $\mu$ P)가 최초로 상품화되어 등장한 것은 불과 5년 전인 1971년이었다. 그 최초의  $\mu$ P는 Intel에서 개발된 4004<sup>1</sup>라고 불리는 16-pin DIP PMOS 4-bit 프로세서로서 45개의 instruction(수행시간이 10.8 $\mu$ sec에서 21.6 $\mu$ sec 사이인)을 가지고 있으며 현재도 쓰이고 있다. 이후 5년간  $\mu$ P 분야는 극적인 성장을 보여 최근의 조사결과 2에 의하면 28개 제조업체에서 무려 37종의  $\mu$ P가 공급되고 있으며, 그 종류도 다양하여 간단한 calculator용 chip, memory도 포함하여 단순 single-chip controller, 4-나지 8-bit data-processor로부터 mini-computer의 CPU 역할을 chip 하나로 할 수 있는 16-bit  $\mu$ P 등 또는 고성능의 특정 제어장치(dedicated controller)나 user-microcode된 CPU의 제작에 쓰이는 bit slice processor 등이 있다. 그 기본 제조 공정은 저(低)전력 고집적도  $\mu$ P의 경우 PMOS, NMOS, CMOS 등이 이용되며, 고속도의 bit slice processor에는 Bipolar, I<sup>2</sup>L이나 ECL 등이 이용되고 있다.

오늘날  $\mu$ P 응용에 주종인 8-bit 프로세서 중에서도 가장 널리 쓰이고 있는 것은 1973년에 나온 소위 제 2세대  $\mu$ P인 Intel 8080<sup>2</sup>로서, 이는 40-pin DIP NMOS 8-bit 프로세서이며 2 $\mu$ sec 내지 9 $\mu$ sec의 cycle time으로 수행하는 78개의 instruction을 가지고 있다. 8080 $\mu$ P는 NMOS 제조 공정에 의해, 1972년도에 제품화된 PMOS  $\mu$ P 8008보다 속도가 10배 빠르게 되었으며, 현재 속도나 온도 특성이 약간씩 다른 몇 종류의 8080  $\mu$ P가 Intel은 물론 AMD, TI, NEC, Siemens, National 등 여러 회사에서 생산되고 있다. 이중 가장 빠른 것은 AMD의 Am 9080A-4로 최단 Instruction 수행시간이 1 $\mu$ sec이다.

8-bit  $\mu$ P로는 이밖에도 12종이 더 있는데 가장 싼 것으로는 National의 SC/MP가 있고, 능력이 가장 뛰어난 것으로는 1976년에 나온 Zilog사의 Z-80<sup>3</sup>가 있다. 후자는 40-pin DIP의 NMOS 8-bit 프로세서로 instruction이 158개이며(최단수행 시간 1.6 $\mu$ sec) 다

른 8-bit  $\mu$ P에서 찾아볼 수 없는 여러가지 뛰어난 특성을 소유하고 있다.

$\mu$ P의 성능의 극적 향상에 따라  $\mu$ P의 가격도 급격히 떨어지고 있다. 예를 들면 8080 $\mu$ P의 경우 TI의 TMS-8080는 1975년말 가격으로 \$34(1~24개)에서 \$21(100개 이상)로 나와 있는데 이는 1973년도 초기가격의 1/10 정도에 불과한 것이다. Z-80의 현재 가격은 \$200(1개당), \$150(2~5), \$50(500개이상)이며<sup>5</sup>, National의 SC/MP는 소대가격이 개당 \$20 정도이다.

돌이켜보면  $\mu$ P는 디지털 집적회로(IC)의 발전 과정상 당연한 귀결인 듯 싶다. 고집적회로(LSI)의 복잡도가 증가함에 따라 생산되는 IC는 대량으로 사용될 수 있는 다목적용이어야 함은 경제적 측면에서만 보아도 자명한 사실이다. 이러한 범용성은 그 기능이 생산 단계에서 고정되어지는 TTL SSI나 MSI와 같은 보다 낮은 집적도의 종래의 IC들과 현저한 대조를 이룬다. 더우기 외부회로와 통신 경로가 되는(package의) pin 수효가 실질적으로 제한되므로<sup>6</sup> 이러한 다목적 회로는 사용 경우에 따라 주어진 일련의 instruction들을 decode하고 execute(수행)하도록 설계되어야 하므로 즉 디지털 컴퓨터의 CPU와 흡사한 기능을 가지게 된다. 디지털 컴퓨터의 경우와 마찬가지로  $\mu$ P를 어떤 특정한 업무에 적용하려면 일련의 instruction들을 프로그램하여 memory에 저장시켜 이루어진다.

$\mu$ P의 집적도와 개발에 필요한 노력의 예로 Intel이 8080 $\mu$ P 설계에 18 man-month를 소비했으며, 약 5,000개의 MOS transistor를 4.2mm×4.8mm 크기의 chip에 넣었다. 집적도의 또 다른 예로 Data General의 micro NOVA, mN 601  $\mu$ P를 보면 종래의 hardwired random logic 방식으로 만든 NOVA840의 CPU는 200개 이상의 SSI와 MSI TTL IC로 구성되어 있고 2장의 15×15 inch 기판을 쓰고 있는 반면 mN601  $\mu$ P는 40-pin DIP으로 package된 크기 5.5mm×6.0mm chip에 NOVA840 CPU의 모든 기능 이외에 stack operation 같은 개선된 성능까지를 포함하고 있다. (다만 mN601은 NOVA840 CPU에 비하여 연산속도가 느린 단점이 있다. 예로 전자의 ADD time은 2.4

正會員 : 韓國科學院教授(工博)

μsec인데 비해 후자의 경우 0.8μsec이다)

디지털 시스템의 제작 비용을 평가하는 대체적이거나 간단한 방법으로 IC의 수효를 세면 된다. 고정된 기능의 IC에 의한 종래의 hardwired logic 설계와 read-only memory(ROM)을 써서 μP를 프로그래밍하는 새로운 방법의 경제성을 비교함에 있어 다음과 같은 주장이 있다<sup>8</sup>. 즉 8내지 16 bit의 memory가 logic gate 하나에 해당되고 또 종래 IC의 평균 gate 수효를 10이라 가정하면 기억용량이 8,000bit인 8K ROM은 50 내지 100개의 IC를 대치할 수 있으며 16K ROM은 100 내지 200개의 IC를 대치 할 수 있다는 것이다. 따라서 종래 방법을 사용한 시스템 제작 경비를 IC당 \$1.60로 잡고 ROM의 bit당 가격을 \$0.02로 치면 50 내지 200개 IC로 구성된 시스템을 고려할 때 새로운 방법은 한 개 시스템당 \$50 내지 \$300의 원가 절감을 가져온다는 것이다. 여기서는 필요한 software나 제반 시스템 개발 비용 등의 경비가 무시되고 제안 방법의 단순화되어 어떤 설계의 경우 위와 같은 결과와는 거리가 있을 것이나 μP를 이용하는 새로운 디지털 설계 방법의 경제적 이점은 명백하다 하겠다.

경제성 이외에도 종래의 hardwired random logic 설계에 비해 μP를 program 하는 새로운 방법의 본질적 이점으로 (1) 유연성(flexibility)과 확장성(expansion)—program은 hardwired logic보다 훨씬 더 변경, 수정하고 확장하기 쉬우므로, (2) 소모 부품 수효의 감소에 따른 신뢰도(reliability)의 증가와 (3) 크기와 소모 전력의 감소 등을 들 수 있다.

따라서 μP는 디지털 설계의 기본 방식을 근본적으로 혁신하고 있다. 더욱 LSI 기술의 급속한 발전에 따른 가격의 급격한 감소에 촉진되어 μP 응용 범위 자체가 가속도적으로 넓어지고 있다. 측정범위를 자동선정(autoranging)하는 digital volt meter, microwave oven 조정장치, POS terminal 등과 같이 종래의 여러 가지 "dumb" 제품에 경제적으로 지혜(intelligence)를 부여하는 데 응용 적합하며 특정 I/O 제어장치, 공정제어, intelligent terminal, 사설교환장치(PBX), 분산 자료처리(distributed data processor)로부터 저렴한 다목적 컴퓨터 시스템(마이크로 컴퓨터) 등에 μP가 응용되고 있다. 특히 마이크로 컴퓨터 시스템의 등장은 처음으로 많은 일반인의 구매능력 이내인 개인 컴퓨터를 가능하게 하였다. 오늘날 16K의 memory와 keyboard, cassette, television interface 등으로 구성된 마이크로 컴퓨터 시스템은 \$2K 미만의 가격으로 구입할 수 있게 되었다.

이런 시스템을 확장시켜 floppy disk나 저렴한 pri-

nter 등의 주변장치를 더하려면 총경비는 현저히 증가 되는데 이는 불행히도 주변장치에 필요한 기계적 장치의 가격이 CPU나 memory 같은 전기적 장치와 보조를 맞추지 못했기 때문이다.

LSI 기술의 앞으로의 경향은 여러 기사에서 논의되고 있다<sup>9</sup>. 과거 15년 사이에 silicon IC의 부품집적도는 1,000배로 증가되었고 CPU의 gate당과 memory의 bit당 가격도 이것에 비해하여 감소 되었다. 특히 dynamic MOS random access read/write memory (RAM)의 집적도는 1972년 이래 매년 2배로 증가되어 1976년말 전으로 bit당 \$0.12(OEM 시장)의 가격의 16K RAM이 생산될 것이고 이러한 경향은 1980년도까지는 계속되어 bit당 \$0.04로 64K RAM이 등장할 전망이다. μP 기술의 경향은 전력 소모와 package pin 수효에 관계되는 문제점 때문에 서로 다른 두 방향으로 발전되어 나갈 것이 예상된다. 하나는 연산속도를 높이는 방향이고 또 하나는 집적도를 높이는 방향으로 1980년 까지는 1초에 적어도 천만개의 instruction을 수행할 수 있는 16-bit μP의 등장이 가능할 것이며 고집적도를 강조한 μP로는 32K의 memory와 여러 I/O interface를 한 chip에 포함하는 16-bit μP가 예상된다. 이러한 LSI IC의 생산 가격은 대량으로 개당 \$10 미만인 될 것으로 추측된다.

μP의 기술적 면모는 이 정도로 살펴보고 다음은 μP 응용과정인 programming과 이에 필요한 software support에 관하여 살펴본다. 종래의 디지털 설계방법과는 달리 μP를 응용하는 방법에는 programming이 개발 작업의 필수적이고 실질적인 부분을 이룬다. 이러한 응용 program을 만드는 데 보조역할을 하는 program, 즉 Support software는 여러 형태로 μP제조 회사나 독립된 software 회사에서 제공되고 있다. 그 중에는 (1) assembly language로 쓰여진 symbolic form의 "source" program을 memory에 담겨질 program의 영상에 거의 일치하는 형태인 μP instruction으로 구성된 "object" program으로 바꾸어 주는 assembler, (2) 보다더 고급 언어로 쓰여진 program을 object이나 assembly language program으로 번역하는 compiler, (3) source program을 쓰고 또한 편집하기 위한 text-editor등 그외 여러가지가 있다.

위와같은 supporting software는 어떤 computer에서 동작하는가에 따라서 cross와 resident의 두 가지로 나눌 수 있다. 전자, 즉 cross-assembler나 cross-compiler는 여러가지로 편리한 시설을 갖춘 중이상대 규모의 "host" 컴퓨터에서 동작하게 되어 있고 반면 resident한 종류의 support software는 실제 응용될

$\mu$ P와 같은  $\mu$ P로 구성된 microcomputer development system에서 동작하게 된다. 후자의 전반적 전산 능력은 전자에 비해 상당한 제한이 있으나 응용 hardware를 연결하여 on-line으로 사용할 수 있고 또한 In-circuit-emulator나 ROM-simulator<sup>10</sup>등 특별한 장치를 이용할 수 있어  $\mu$ P응용에 필수적인 항목이 된다.

Cross software는 time-sharing computer center를 이용하여 사용하는 방법외에 직접 (보통 ANSI FORTTRAN IV로 쓰여진) source form으로 구입하여 FORTRAN compiler를 쓸 수 있는 어떤 computer에서나 사용할 수 있다. (단 이런 host computer의 word 길이가 32 bit 이상이어야만 작동할 수 있는 경우가 많다)

이와같은 support software는 그 능력의 차이에 따라 또 구별할 수 있다. Assembler의 경우 symbolic addressing이 불가능하고 한 line씩 assemble하는 간단한 incremental assembler (resident 종류)로부터 강력한 macro 작용이 가능한 macro assembler<sup>11</sup>까지 있으며 bit slice processor를 사용하는데 필요한 microprogramming에는 또 다른 특성을 가진 assembler<sup>12</sup>가 쓰여진다.

$\mu$ P를 위하여 특별히 만들어진 compiler는 현재 수효가 적다. 1975년 말까지 유일하게 알려진  $\mu$ P compiler로써 Intel의 8080과 8008을 위한 PL/M<sup>13</sup>이 있고(syntax는 IBM의 PL/I 특히 PL/I의 부분을 딴 XPL<sup>14</sup>과 같음) 그후 Motorola, 6800을 위한 PL/M 6800, Signetics Z6S0을 위한 PL $\mu$ S, Zilog Z-80를 위한 PL/Z가 등장하였다. 이상 compiler는 모두 crosscompiler이며 아직까지는 8-bit  $\mu$ P용 resident compiler는 널리 알려진 것이 없다.

DEC에서 제품화된 LSI-11, Data General의 mN601과 TI의 TMS9900은 각기 이들 회사에서 전부터 생산되던 mini computer의 CPU를 모방한 16-bit  $\mu$ P 이미로 전에 mini computer system software로 개발되었던 FORTRAN, COBOL 혹은 ALGOL compiler 등 또는 BASIC이나 이와 유사한 interactive programming 언어를 이용할 수 있다.

응용 program을 쓰기 위한 이상과 같은 translator의 효과를 다음과 같이 평가된 바 있다<sup>9</sup>. Program 개발에 드는 경비는 documentation, debugging, 또한 앞으로의 maintenance 경비를 포함하여 (1) translator를 사용하지 않고 손으로만 할 때 program bit당 \$5(물론 이런 방법은 프로그램 길이의 100 byte 정도 넘으면 실질적이지 않음), (2) assembler를 이용하면 bit당 \$1, (3) compiler를 이용하는 경우 \$0.20이라

고 추정된 바 있다.

편리성과 경제성의에도 compiler를 사용하여 high level 언어로 program을 만들면 또 다른 이점이 생긴다. 즉 program의 이동가능성(trans portability)이다.  $\mu$ P의 종류를 변해도 program을 다시 개발할 필요가 없어진다. 새로운  $\mu$ P에 다시 compile하면 되기 때문이다<sup>15</sup>.

현재 쓰이고 있는  $\mu$ P compiler에 관하여 compile된 program의 memory면에서의 비효율성이 문제시되고 있다. (보통의 MOS LSI  $\mu$ P응용에서는 time은 덜 중요하다). 그러나 memory의 가격은 계속하여 하락되는 반면 soft ware 경비는 상승하고 있으므로 이러한 비효율성은 무시되는 경향이며 compiler 자체가 효율적으로 개발되고 있으므로 (예로는 IBM의 optimizing PL/I<sup>16</sup>) 앞으로는  $\mu$ P응용에도 compiler가 가장 중요한 자리를 차지할 것이다. 이런 관점에서 볼때 assembler와 compiler의 중간격인 PL/360<sup>17</sup>와 같은 translator가  $\mu$ P응용에 더 적합하게 보이나 아직까지는  $\mu$ P를 위한 이와같은 software가 널리 알려진 것이 없다.

Assembler, compiler 또는 text-editor외에 micro computer development system은 resident software로서 loader, I/O handler등 필요한 monitor program이 있어야 한다. (이런 program은 주로 ROM에 설치되어 있는 "firmware"로 마련된다.) 이외에도 program이 틀린 곳을 찾는 데 도움이 되는 debugging program 또는 hardware를 test 하는데 필요한 diagnostic program이 쓰여지고 있으며 micro computer system의 능력이 확장됨에 따라 system software도 더 필요하게 된다. 예를들어 보조기억장치로 floppy disk를 사용하려면 disk operating software가 있어야 되며, PROM을 program하는 장치, in-circuit-emulator, 또는 ROM simulator와 같은 on-line 장치를 동작시키는데 필요한 software가 요구된다.

어떤 응용 program의 assembly나 compile 과정이 error가 없이 수행되었다고 해도 물론 그 program의 타당성(validity)은 보장되지 않는다. 여러가지 조건하에서 program을 실제로 사용해 보는 방법외에는 실용적이고 효과적인 방법이 아직까지는 알려져 있지 않다. 이런 실험적인 test는 simulator/interpreter를 사용하여 응용 hardware를 쓰지 않고도 어느정도 어떤 "host" computer에서 시행할 수 있다. 이런 방법을 사용하면 아래와 같은 이점이 생긴다. (1) 응용 hardware가 완성되기 전에 program을 개발할 수 있다. (2) 이와같은 독립적으로 debug된 program은 완성된

hardware를 test 하는데 유용하다. (3) 응용에 따른  $\mu P$ 선정을 system을 실제로 만들어 보기전에 어느정도 확실히 평가할 수 있다.

이러한 simulation 방법은 특히 micro code된 CPU의 설계등 micro coding 작업에 본질적으로 필요하다.

새롭고 여러모로 더욱 향상된  $\mu P$ 가 과거에 상상외로 속히 개발되는 점<sup>18</sup>을 고려하면 필요한 system software (assembler, compiler, simulator등)를 개발하는 방법도 종래의 hand coding과는 다른 새로운 방법이 발견되어야 한다는 것이 분명하다. 이런 목적의 연구가 필자에 의하여 수행되어 1975년초에는 MAC74<sup>19</sup>라고 불리는 processor를 이용하여 반자동적인 방법으로 임의의  $\mu P$ 를 위한 cross assembler와 simulator를 IBM OS assembly language로 제조하는데 성공하였다. 자세한 내용은 다음 기회에 논하기로 한다.

끝으로  $\mu P$ 응용의 교육적인 환경을 고려해 보기로 한다.  $\mu P$ 응용의 연구, 개발 및 교육에 필요한 실험실을 한국과학원에 설치하려는 과정으로 과학원의 NOVA 840 mini computer에 8080 $\mu P$ 를 위한 assembler/simulator인 A8076/P8076을 만들었다. 이 software는 과학원 응수와 1학년 학생들에게 assembly language programming을 가르키는데 이용되었고 또 동시에 이 software 자체가 debug 되었다. program을 P8076을 사용하여 run 할때마다 소요된 memory byte양과 cycle time이 정확히 측정됨으로 학생들의 program 기술 향상을 쉽게 알 수 있었으며 한학기가 끝날무렵 학생들은 Tarjan의 algorithm<sup>20</sup>과 같은 어려운 program을 8080  $\mu P$ 에 약 500 byte 내로 쓰는데 성공하였다. A8076/P8076과 같은 software가 3000series bit slice processor와 SC/MP  $\mu P$ 를 위하여도 NOVA 840에 개발되고 있는 중이다.

결론으로 LSI  $\mu P$ 는 기술한 바와 같이 디지털 설계 방법에 큰 영향을 주고 있다. 경제성이나 편리성만 고려해도 종래의 hardwired logic design은 범용적 LSI processor를 program하는 새로운 방법으로 점차 대체되고 있는 것이 타당하여  $\mu P$ 와 또 관련된 LSI 제품의 성능향상 이에 비례하는 가격의 감소로  $\mu P$ 의 응용은 급속히 확대되어 가고 있는것이 분명히 보인다.

디지털 컴퓨터의 발명은 언어의 발명과 조직적인 과학의 발명에 이어 인류의 지적역사의 세번째 거대한 단계를 이루었다고 논했다<sup>21</sup>. LSI  $\mu P$ 발명은 멀지않아 일반 가정용 디지털 컴퓨터를 가능하게 할 것이며 이와 같은  $\mu P$ 응용의 급속한 발전 경향은 앞으로 여러해 계속하리라고 기대되고 있다.

### 참 고 문 헌

1. MCS-4 Microcomputer Set Users Manual, Intel (1974).
2. "Microprocessors", Electronics, 49, 8 (1976) 74-174.
3. Intel 8080 Microcomputer System Users Manual, Intel (1975).
4. 280-CPU Technical Manual, Zilog (1976)
5. Private Communications with Zilog Staff in 1976.
6. 현재 최대 pin count는 8-bit  $\mu P$  경우 40이며 16-bit  $\mu P$  경우 64이다.
7. Cushman, R.H., The Intel 8080: First of the Second Generation Microprocessors, EDN 19, 9 (1974) 30-36.
8. At one of Intel Microcomputer Systems Seminar in 1975.
9. Hodges, D.A., Trends in Computer Hardware Technology, Computer Design (1976) 77-85.
10. For example, MDS-ICE-80 and SIM-101/-102 /-104. Intel Data Catalog (1976).
11. For Illustration of These Features see IBM OS/VS Assembly Language, GC 33-4010-2 IBM (1974). No known Commercial  $\mu P$  Cross Assembler has Features as Varied and Powerful. As Discussed Subsequently the Author has Prooved OS Assembly Language Compatible Cross Assemblers for Several  $\mu P$ s in a Semiautomatic Way.
12. Such as CROMIS of Intel. Intel Series 3,000 Microprogramming Manual, Intel (1976).
13. 8008 and 8080 PL/M Programming Manual, Intel (1975).
14. McKeeman, W.M., et. al., A Compiler Generator, Prentice-Hall (1970).
15. 근본적으로는 어떠한 프로그램 (object도 포함하여)도 이동가능성이 있다. object 프로그램인 경우 disassembly, decompilation과 compilation 과정으로 이동하는 방법이 있는데 현재는 연구와 실험단계를 벗어나지 못하고 있다.
16. OS PL/I Checkout and Optimizing Compilers: Language Reference Manual, GC 33-0009-3, IBM (1974). (p. 에 계속)

- 17. Wirth, N., A Programming Language for the IBM 360, JACM 15 (1968) 37.
- 18. LSI 제품을 개발하는 회사의 견지로는 이런 제품의 life cycle이 5년을 넘지못함으로 새로운 제품이 2~3년전마다 계속하여 나와야 된다고 한다(8 참조)
- 19. Park, J.C.H., MAC 74, A General Purpose Macro Processor, SLAC-175 (1974) Avarilable from NTIS.
- 20. Tarjan, R., Depth-first Search and Linear Graph Algorithms, SIAMJ. Comput. 1,2(1972).
- 21. Gerard, R.W., Computers and the Future of Education, Talk given at 1965 AFIPS Fall Joint Comput. Conf.

2

會員에게 알리는 말씀

會員여러분의 健勝하심을 仰祝합니다.

빛나는 祖國의 繁榮과 보다 잘살기 위한 근면, 자조, 협동의 새마을정신으로 힘찬 전진의 노태가 메아리치고 있는 이때 우리는 더욱 總和團結로서 健全한 社會氣風 造成으로 庶政刷新推進에 더욱 앞장서서 근면, 검소한 生活로 職場에서나 家庭에서나 自己일에 充實하여야 할 것 입니다.

따라서 政府의 庶政刷新推進對策(自體監查嚴正實施, 不合理的 制度改善, 생활주변정화운동, 自體教育強化, 對民啓導活動展開等)을 會員 여러분께서는 가족, 친지, 동료직원 및 모든 사람에게 이같은 사항을 주지 시키시어 밝은 社會建設을 위한 健全한 社會氣風 造成에 殊선수범하여 주시기 바랍니다.

1976年 11月 日

大韓電氣學會長 白

庶政刷新은 좋은 나라를 建設하겠다는

우리社會의 “조용한 精神革命”입니다