最新技術

컴퓨터 설계의 Microprogramming

宋 榮 宰

慶熙大 電子工學科

1. 머리말

Microprogramming의 개념은 Cambridge 대학의 M.V. Wilkes 교수에 의하여 최초로 제안*된 것으로서, 컴퓨터의 Control부를 계통적이며 조직적으로 설계하는 방법을 제공하는 것이 목적이였다[1].

컴퓨터의 program이 명령의 조합으로 구성되어 있는것과 대조적으로 컴퓨터의 명령은 micro 명령의 조합으로 구성되어 있다. 이것을 Microprogram이라 부른다. 즉 컴퓨터의 각 명령은 각각 몇개의 Micro명령의 조합으로 되어있는 Microprogram으로 한다. 여기에서 이 Microprogam을 적당한 기억장치(이것을 Control기억이라 부른다)에 기억시켜 놓고, 그 Micro명령을 순차적으로 읽어내어 그것이 지시하는 Micro조작을 순차적으로 실행함으로써 컴퓨터의 명령을 실행할 수 있게 된다. 이와같은 Control 방식을 Microprogram Control방식이라 부른다.

1964년 I.B.M System/360에 Microprogram Control방식이 대대적으로 채용한 이래, 컴퓨터의 Control 방식은 종래의 Transistor 논리회로에 의한 Hard Wired Control방식(Wired logic)으로부터 Control Storage에 기억시킨 기억논리에 의한 Microprogram 방식으로 옮겨지고 있다. Microprogram Control기술은 근년에 급속히 주목되어 실용화되고 있으며, 오늘날의 Control 논리 설계는 물론, 미래의 컴퓨터에 크게 영향을 미칠것으로 전망되고 있다.

* Wilkes의 1951년 컴퓨터 회의에서 발표한 "The Best Way to Design Automatic Calculating Machine"

2. Microprogramming의 발전

1951년 Wilkes에 의해 Microprogram 방식이 제안된이래 지금까지의 발전과정을 살펴보면, Table 1에 표시한 바와같이 3기로 나누어 생각할 수 있다[2].

Table1 : Historical review of Microprogramming

제 1 기(1951~1964)		
1951	M.V.Wilkes	Microprogramming 방식의 제안
1956	W.Renwick	EDSAC Ⅱ
1958	G.P.Dinneen	CG24
1959	Vandel Poel	ZEBRA
1960	T.W.Kampe	SD-2
1961	H.M.Semarne	Stored Logic의 개념과 AN/UYK-1
1961	H.Isaksson	GIER
1962	A.Grasselli	Pathfinder Memory의 제안
1963	G.B.Gerace	CEP
1963	M.W.Allen	CIRRUS
1964	W.C.McGee	TRW-133

1964	E.O.Boutwell	PB-440
1964	L.Beck	C-8041
제 2 기 (1964~1969)		
1964	IBM	System/360
1965	B.R.S.Buckingham	CAS
1965	S.G.Tucker	Emulation
1967	A.Opler	Firmware의 제한
1967	H.Weber	Microprogrammed Implementation of EULER
1967	G.A.Rose	Intergraphic
제 3 기 (1970~)		
1970	N.Bartow	Microdiagnostics
1970	M.J.Flynn	Dynamic Microprocessor
1970	C.V.Ramamoorthy	User-Microprogramable Computer
1970	S.S.Husson	"Microprogramming: Principle and Practices"
1971	A.B.Tucker	dynamic Microprogramming.
1971	R.H.Eckhouse	MPL
1971	R.Zacks	A.Firmware APL time-sharing system.
1972	E.W.Reigel	The Interpreter
1972	R.F.Rosin	QM-1
1972	W.T.Wilner	B-1700
1972	Micro-5	Firmware Monitoring.

즉, 제 1 기는 1951년부터 1964년까지의 기간으로써 Microprogramming에 대한 여러가지의 아이디어가 나왔고, 각종 컴퓨터가 만드러졌던 시대이다.

제 2 기는 IBM System/360의 고정기억(Read only storage)에 Microprogramming방식을 도입한 Control 방식이 컴퓨터에 채택되었던 시기이다.

제 3 기는 Writable한 Control 기억이 실용화되어 Firmware가 활용하게끔 되었던 시기라고 말할 수 있다.

3. Microprogramming의 특징

설계라는 의미를 system 설계를 포함해, 넓은 의미로 해석하고, Microprogramming의 특징을 살펴보면 다음과 같다.

우선 장점으로 생각되는 것은,

(1) 설계의 간명화 : Control부가 통일적으로 되고, 설계기간의 단축이 가능해진다.

(2) 분업화, 표준화가 용이한 점 ; 일반적으로 컴퓨터의 Control부는 설계가 가장 복잡하고 어려운 부분으로써 설계자에 따라서 개인차가 나는 곳이지만, Microprogramming Control에 의한 간명함과 일관성에 의해 설계의 분단화, 표준화가 용이하다.

(3) 보수, 진단이 용이한 점 ; 설계의 규칙성으로 부터의 당연한 귀결로서 Micro진단 program에 의해서 세밀한 Micro 진단을 가능케 한다.

(4) Hardware 설계의 교육이 용이한 점 ; 종래의 복잡한 Control부분이 간명해지므로 Hardware설계자, 보수자에 대한 교육이 용이해진다.

(5) 낮은 코스트로써 풍부한 명령기능을 실현가능케 한다 ; 적은 Hardware Resource에도, 속도적으로는 떨어지지만 microprogramming으로써 복잡한 기능을 가진 명령 set를 실현시켜, 컴퓨터의 Family화를 용이하게 할 수 있다.

(6) 설계, 제작기간의 단축 ; Hardware의 세부를 Microprogramming으로 Control할수 있기 때문에 기능의 변경, 추가가 용이하고 Hardware 세부의 논리설계가 정해지지 않아도 Hardware의 설계, 제작이 가능하다. 따라서 종합적인 제품화의 기간을 단축시킬 수 있다.

(7) System의 수명이 연장된다 ; Microprogarmming Control이 지닌 유연성으로 인하여

System Architecture의 변경을 어느정도 Microprogramming에 의하여 흡수 할 수 있다.

(8) LSI화에 적합 ; Microprogramming Control방식에 의해서 Control부와 연산부가 논리적으로나 물리적으로 분리할 수 있다. 그러나 LSI화가 곤란한 Control 회로의 대부분을 Microprogram속에 흡수할 수 있는점을 고려에 넣으면 계산기의 LSI화에 적합한 방식이라 할수 있다.

(9) Emulation이 용이하다 ; 기존 Program의 활용이 용이해지므로 새로운 시스템의 개발이 편리해진다.

한편, 결점으로서는

(1) 조작효율의 손실 ; Microprogramming에 대한 기대란 유연성을 얻을 수 있는데 있지만, 그것이 System Programer에게도, 조작원에게도 또 컴퓨터를 설치하는 경우 관리면에서도 많은 문제를 내포할 가능성이 있다.

(2) 성능의 손실 ; 처리 속도가 Microprogram 기억의 동작속도에 관계되므로 고속 컴퓨터에는 적합하지 않다. 그러므로 대형기에서는 고속처리가 요구되는 일반적인 기본명령은 배선논리로써 실현시키고, 많은 논리조작을 필요로하는 명령과 진단기능등은 Microprogram 논리에 의해 Firmware화시키는 경향이 있다.

(3) Cost 면에서의 결점 ; Architecture가 간단하고 고정화되어 있는 System과 System기능이 비교적 간단하고 반복되는 작업에 한정되어 있는 System의 경우, Microprogramming은 매우 고가가 된다.

(4) Microprogramming을 하기 위해서는 약간의 Hardware의 세부를 이해할 필요가 있고, 고급언어레벨의 사용자에게는 곤란이 많다.

4. Microprogramming의 종류

Microprogramming Control의 개념은 다음의 두가지로 분류할 수 있다.

(1) 수평형 Microprogramming Control; 이것을 Direct Control방식이라고도 한다. 이 방식은 Control word의 각 Bit가 System중의 각 Control Gate와 1대 1로 대응되게끔 한 Control 방식이다. Control Gate의 수가 적은 소규모의 System에 적합하다. 이 방식의 특징을 아래에 표시한다.

a) Control기억의 Capacity가 커진다.

b) 확장성이 부족하다.

c) Hardware의 세부의 Program Control이 가능하다.

d) Microprogramming이 어렵다.

e) 고장진단이 용이하다.

f) 병렬처리가 많고 속도가 빠르다.

g) Decoder가 필요없으므로 해독에 의한 시간지연이 없다.

h) 순서 Control에 의한 Address선택은 직접지정방식을 취한다.

(2) 수직형 Microprogramming Control; Encode Control 방식이라고도 하는데 이 방식은 System속의 Control Gate를 개폐하는 Control 신호가운데 동시에 작용해서는 안되는 것, 동시에 작용할 필요가 없는 것, 즉 서로 배타적인것을 group화해서 예를 들면 동일 group에 속하는 Control신호가 M개 있는 경우에는 $M \leq 2^m - 1$ (m,M는 정의 정수)를 만족하는 최소의 정수 m bit를 갖는 Control field를 M개의 Control Gate용에 ROS (Read Only Storage)에 나누어 Coding한 Control방식이다. 이 방식은 소형이나 Mini Computer 규모의 컴퓨터에 많이 채용

되고 있다. Micro 명령의 형식이 일반적인 기계명령에 가깝다. 수직방식의 특징은 다음과 같다.

a) 순서 Control은 Microprogramming Counter에 의한 순서 Control이다.

b) Control word의 처리가 순차적이므로 처리효율이 나쁘다.

c) Microprogramming이 용이하다.

d) Control기억의 Capacity가 적게된다.

e) Control word는 수종류의 Field Type를 가진다.

f) 수평형과 반대로 Decoder가 필요하므로 해독하는데 의한 시간의 지연이 생긴다.

g) 확장성이 풍부하다.

여기에서 지금까지 이미 발표된 Microprogramming Control 컴퓨터의 Control word 구성예를 참고로 표시한다.

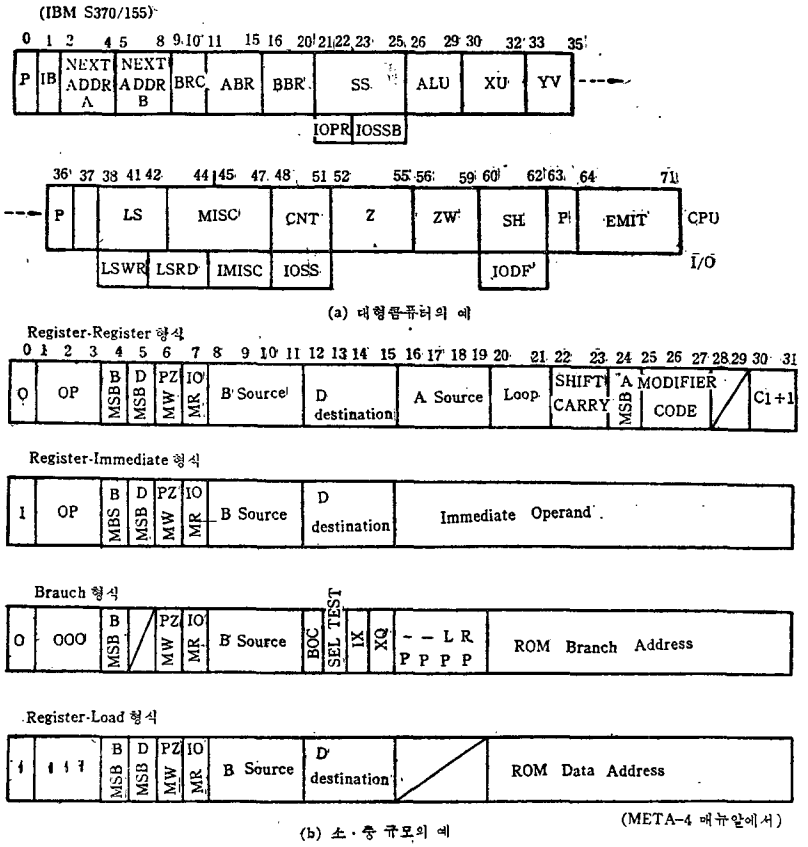

그림 1

5. 새로운 microprogramming기술

microprogramming기술은 상용기계에 채용된 이래 컴퓨터의 system설계 기술로서 확립되고 있다. 한편 그 개념과 기술은 금후에도 발전, 개량시킬 여지가 있다. 요즈음에는 user에게 microprogramming을 개방하는 dynamic micro-

programming, Firmware화하고 있다.

(1) dynamic microprogramming

25~50ns 정도의 고속 writable한 control 기억(Writable Control Storage, WCS)의 출현에 따라 microprogramming 컴퓨터는 새로운 국면을 맞이 하였다고 볼 수 있다. 이와같이 WCS를 이용한 새로운 기술을 dynamic microprogramming이라고 한다.

1951년에 M.V.Wilkes가 microprogramming을 제안했을 당시 벌써 microprogramming의 writable에 대해서 언급되고 있다. 그는 「writable한 control 기억을 사용하면 프로그래머는 자기자신의 요구에 합치된 명령코드를 선택하는 것이 가능하고, 또 필요하면 program(실행중에)의 도중에서 명령코드를 요구하는 것이 가능하게 되리라」 언급하였다.

IBM system/360에서 Emulation이 사용되었고[3], Weber에 의해 고급언어의 microprogrammimg에 대한 처리방식이 발표되어 microprogramming을 변경하는 방식에 대한 관심이 점차 높아졌었다.

1967년에 A.Opler는 Firmwave라는 새로운 술어를 만들어 내서 다시금 microprogram control 컴퓨터의 특징에 눈을 돌리게 되었다[4]. 또 Flynn은 dynamically alterable storage의 사용을 생각해 내서 dynamic microprogramming에 대해서 논하였다[5].

dynamic microprogramming 방식의 컴퓨터는 다음과 같이 소수의 종류에 불과하다.

Meta 4(최초의 DMP 방식의 컴퓨터)

Nanodata QM-1

Burroughs B 1700

Micro data 1600, 3200

Hewlett Packard 21MX

Inter data Model 85, 8/32

Varian 73, 74, 75

(2) Firmware

이 Firmware라는 용어는 Datamation의 1967년 1월호에 게재된 Ascher Opler의 논문(Fourth Generation Software)에서 최초로 사용되었다. 그것은 컴퓨터 system이 Hardware와 Software로서 구성되어 있다고 생각할 때, 그 중간에 Interface라는 것이 기계어명령이라 할수 있다. Hardware와 Software의 접점에 새로운 firmware를 생각하면, program을 구성하는 명령이 Software와 Firmware와의 사이에 Interface가 된다. 이 명령을 Firmware, 즉 microprogramming이 해석실행하는 것이 된다. 그리고 Firmware와 Hardware 사이에 Interface로서 micro 명령이 존재하는 것이다.

microprogramming방식의 컴퓨터에서 명령은 Firmware에 의해 결정된다. 따라서 Firmware를 변경함에 따라 Hardware를 변경하지 않고도 어느 정도 자유로이 명령을 바꿀수 있다. 특히 control을 써서 넣을 수 있다면 더욱더 쉽게 명령을 변경할 수 있고, 컴퓨터 system의 융통성 유연성이 현저하게 증대되게 된다.

주로 firmware를 이용한 컴퓨터는 Weber의 EULER컴퓨터[6], Zaks의 APL컴퓨터[7], B-1700 등이 있다.

6. microprogramming의 응용

microprogramming은 앞에서 기술한 바와 같이 특징과 이점으로 인하여 현재의 컴퓨터에서는 중요한 기술이 되고 있다. 컴퓨터뿐만 아니라 여러가지의 응용분야에 걸쳐서 각종 논리장치에 적용되어 앞으로도 더욱더 응용분야가 넓

혀지리라 본다.

microprogramming의 응용은 다음과 같이 분류할 수 있다[8].

(1) 충분한 Architecture를 가진 System에 Control의 대한 효율이 좋은 설계가 가능하다.

(2) 기존 Computer Architecture의 일부 변경에 사용된다.

(3) 어떤 기존 Hardware 능력의 Emulation에 매우 유효하다.

(4) Hardware와 여러가지 레벨의 Software (OS로 부터 응용 program까지)를 결합해서 일관성 있는 전체 System을 설계하는 일이다.

(5) 특수명령, 확장명령의 부가 등이다. 이 밖에 중요한 응용분야로서 microprocessor 및 Computer Complex System의 분야가 주목된다.

7. 금후의 과제

Microprogramming의 실용화는 수년 사이에 급속히 발전하고 있다. 여기에서는 컴퓨터설계라는 관점으로 부터 금후에 남아있는 과제로 지목되고 있는 몇가지의 문제를 소개하겠다.

(1) micro 명령

새로운 컴퓨터의 개발에 즈음하여, micro 명령이 지녀야 할 기능의 설정은 중요한 문제로 되고 있다. 이 설정은 개발하고자 하는 컴퓨터의 목적에 따라 다소 달라지게 된다. 특정한 기계명령을 가질 경우에는 그것에 맞는 기능을 준비하면 좋지만, Emulation을 주목적으로 한 컴퓨터(MLP-900, QM-1 등)에서는 기계적인 완전성(functional completeness)을 배려해야만 한다.

micro 명령의 형식에서 논리설계가 쉬운 것은 수직형 쪽이라 하지만 처리속도의 관점에서는 병렬조작이 가능한 수평형이 유리한 것으로 평가되고 있다. 논리설계의 전문가에게는 후자가 좋지만 microprogramming 기억용량, bit 이용율등에서는 많은 문제가 있다. 현재로서는 수직형 또는 이것에 가까운 micro명령을 설정하는 것이 일반적이다. 처리속도의 개선은 반도체 기술의 진보에 전적으로 의존하고 있는 형편이다. 결국 경제적 측면에서 어디까지나 수평형에 가까와 질까하는 하나의 기술적인 과제라고 보고 있다.

대형고속컴퓨터에 있어서의 병렬처리 pipeline 처리 및 연상처리 Control을 위해서 어떤 micro 조작을 준비할까하는 문제는 금후의 중요한 과제로 되고 있다.

(2) 동적 microprogramming

당연한 귀결로서, 반도체기술의 진보에 따라 user에 microprogramming 방식이 상식화 하리라 생각되지만 이 경우는 Hardware 기술 이외에 microprogramming을 사용자에게 공개하는 방법이 문제가 된다. 우선 첫째로, 일반 사용자가 쉽게 firmware를 만들수 있도록 microprogramming Assembler와 B-1700에서 보는바와 같이 MIL(Microprogram Implementation Language), SDL(System Developemnt Language) 및 Simulator등을 개발하는것이 급선무로 지적되고 있다. 이와같은 것은 user program의 처리효율을 생각해서도 말할 수 있는 것으로서 장래는 지금까지와 같은 기계명령레벨의 중간언어를 가지지 않고도 micro 명령으로 부터 직접 명령으로 변환시키는 것이 많을 것으로 생각되고 있다.

(3) Hardware 구성

IBM System/360, microcomputer등 microprogram 컴퓨터에서는 Register군을 공통 BUS를

통해 접촉하여 구성하는 것이 많다. 이것은 sub unit의 표준화, LSI화에 연결되고, 또한 기능의 확장, 실장등에서 특징을 살린다. 또 고장진단, 보수가 쉬워지는 이점이 나온다. 따라서 앞으로도 BUS구조방식의 채용이 상식화되리라 생각하는 경향이지만, data bus가 길어지고 처리속도가 떨어지는 결점과 LSI화의 경우에 접속점이 증가할 염려가 있다.

(4) 입출력 Control

CPU의 Control 회로에는 microprogramming 방식이 완전히 적용되어 그 유용성도 이해되고 있으나, microprogramming의 장점을 입출력 Control장치에도 충분 살리는 것이 유효한 과제로 보고있다, 현재의 입출력장치에는 적극적으로 활용되지 못한 이유는 입출력 Control방식이 CPU의 Control방식처럼 기능적으로 표준화되어 있지 않고, 용도별로 특별한 Control장치를 개발했기 때문이다. 따라서 microprogramming이 가진 범용적인 기능으로 부터 생각해도 앞으로는 범용 microprogramming 입출력 Control장치가 활용되리라 보고 있다.

입출력 Interface의 표준화에 대해서도 똑같다고 본다. 즉 적당한 신호선을 준비해서 신호의 의미를 microprogramming으로 해석해서 다양성이 풍부한 입출력기기의 Control을 통일되도록 하는 것이 바람직하다고 보는 경향이다.

8. 맺는 말

microprogramming에 대해서 컴퓨터의 설계라는 폭넓은 관점에서 살펴 보았다.

microprogramming은 완전히 실용단계에 들어섰고, 많은 문제점이 지적되고 있다. 앞으로도 microprogramming을 둘러싼 흥미있는 과제도 다양하게 제기될것으로 생각되며, 또한 기대된다.

참 고 문 헌

(1) Samir S. Husson: Microprogramming; Principles and Practices, Prentice-Hall, 1970.

(2) M.V.Wikes: The growth of interest in microprogramming, Computing Surveys, Vol. 1, No. 3, pp. 139-145, Sept., 1969.

(3) S.G.Tucker: Emulation of large systems, Comm. ACM, Vol.8, No. 12, pp. 753~761, Dec., 1965.

(4) A.Opler: Fourth generation software, Datamation, Vol.13, No. 1, pp. 22-24, Jan., 1967.

(5) M.J.Flynn al.: Dynamic microprogramming Processor organization and programming, Comm. ACM, Vol.14, No. 4, pp. 240-250, 1971.

(6) H. Weber: A microprogramed implementation of EULER on IBM System/360 Model 30, Comm. ACM, Vol.10, No. 9, pp. 549-558, Sept. 1967.

(7) R.Zaks: Afirmware APL time-sharing system, Proc. SJCC, Vol.38, pp. 179-190, 1970.

(8) J.A.Clapp: application of microprogramming technology, Signicro Newsle Vol.3, No. 1, pp. 8-47, Jan.,1972.