

마이크로프로세서의 발전

맹승렬*, 이종민**

(*한국과학기술원 전산학과 교수, **한국과학기술원 전산학과 박사과정)

1. 서론

1971년 인텔에서 세계 최초로 4004 마이크로프로세서를 개발한 이래 반도체 기술의 발전과 더불어 마이크로프로세서는 급격하게 발전하여 우리 삶에 많은 영향을 미쳐 왔으며 앞으로도 세상을 더욱 변화시킬 것으로 기대된다. 마이크로프로세서는 기술 발전과 더불어 보다 낮은 경비로 보다 우수한 성능을 가지게 되면서 개인용 컴퓨터(PC; Personal Computer)를 가능하게 하였으며, 이러한 컴퓨터를 사용하여 기존의 문서를 전산화한 종이 없는 사무실, 컴퓨터 단층촬영(CT)이나 영화를 더욱 실감나게 해주는 화상처리와 동영상, 자동차 또는 비행기 제작시 컴퓨터 시뮬레이션을 통한 보다 안전한 설계, 암호화 기술을 사용한 사생활 보호 등이 가능하게 되어 생활 주변에 두루 사용되게 되었다 [1, 2].

이러한 마이크로프로세서의 발전 단계를 연대별로 정리하면 다음과 같다[3]. 1946-1970년대는 마이크로프로세서가 등장하기 전의 시대로 1946년에 J. P. Eckert와 J. Mauchly가 진공관을 사용하여 ENIAC이란 컴퓨터를 개발하여 컴퓨터 산업이 생겨나게 된 시기이며, 또한 1948년의 트랜지스터 발명과 50년대 후반의 집적회로 개발은 이후의 CPU 발달에 촉매가 된다. 1971-1973년 기간은 CPU 개발의 시작 단계로서, 1971년에 인텔에서 ENIAC의 성능에 필적하는 4004 마이크로프로세서가 처음으로 개발되었으며 모토롤라, 텍사스 인스트루먼트 같은 회사에서 CPU 개발을 준비하는 단계였다. 1974-1979년 기간은 6800, 8080, Z80 같은 8비트 CPU의 개발과 더불어 PC가 등장하였으며, 1978, 1979년에는 16-비트 CPU인 인텔의 8086과 모토롤라의 68000이 등장하여 각각 IBM PC와 애플사의 매킨토시에 채택되어 본격적인 마이크로프로세서 패밀리의 기초가 된다.

1980년대는 마이크로프로세서 발전이 성장기로 접어들다. 1980-1985년 기간에는 인텔에서 80286, 모토롤라에서 68020을 개발하여 상용 마이크로프로세서의 호환성을 기반으로 성능 향상을 추구하는 시기였다. 또한 기존의 CISC(Complex Instruction Set Computer) 구조를 탈피하여 보다 나은 성

능을 가지는 RISC(Reduced Instruction Set Computer) 구조의 마이크로프로세서를 개발하기 위한 Stanford대학의 MIPS 프로젝트, U.C. Berkeley의 RISC 프로젝트가 진행되어 상용화되었으며[4,5], 이후 CISC 마이크로프로세서와 한 축을 이루면서 발전하게 된다. 1986-1991년 기간에는 인텔의 x86 계열 CPU와 모토롤라의 68000 계열 CPU가 업계를 주도하였으며, 호환성을 유지하면서 성능 향상을 위한 클럭 속도 증가, 캐쉬 메모리의 사용, 파이프라이닝 기법 적용, 분기 예상 기법(branch prediction), 수학계산을 위한 코프로세서 사용 등의 여러 기법이 사용되었다. 이에 따라서 모토롤라에서 68030, 68040을, 인텔에서는 80386과 80486을 잇따라 개발하였다. 그러나 기존의 한 사이클에 하나의 명령어를 수행하는 스칼라(scalar) 구조의 CPU로는 더이상 성능 향상이 어려워지게 되어 새로운 CPU 구조를 필요로 하게 되었다.

이에 따라 1992년 이후에는 동시에 여러 개의 명령어를 수행할 수 있는 슈퍼스칼라(superscalar) 구조와 기존의 파이프라인 단계를 보다 세분화한 슈퍼파이프라인(superpipeline) 기법을 채택하여 성능 향상을 추구하여 인텔의 Pentium, DEC 21064, 애플-IBM-모토롤라 진영의 PowerPC 같은 마이크로프로세서가 등장하였으며, 점점 더 고성능화되어 가고 있는 추세이다. 이는 현재의 마이크로프로세서 설계에 있어서 중요한 고려사항이 되고 있다. 본 고에서는 기술 발전에 따른 마이크로프로세서 구조에 대하여 고찰함으로써 앞으로의 마이크로프로세서 발전 추세에 대하여 살펴보고자 한다. 2절에서는 마이크로프로세서 구조의 연구 방향에 대하여 살펴보고, 3절에서는 이를 응용한 산업계의 마이크로프로세서 발전 추세에 대하여 고찰한다. 마지막으로 4절에서는 마이크로프로세서 구조 발전에 대하여 요약한다.

2. 마이크로프로세서 구조의 발전

2.1 RISC 구조

1970년대에 CPU가 단일 칩에 들어가는 마이크로프로세서화되면서 성능 향상을 위하여 점점 높은 기능의 연산자

들을 명령어 집합에 포함시켜 사용자의 편의를 도모하며 프로그램의 수행시간을 단축시키는 CISC 구조 [6]가 보편화되었으나, 이로 인하여 제어기가 복잡해져 CPU 클럭의 증가가 어려워져 속도의 한계를 가져왔다. 이에 반하여 1970년대 말과 1980년대 초에 걸쳐 그 당시의 기술로 32-비트 컴퓨터를 만들기 위한 연구가 진행되어 기존의 CISC 구조에 비하여 명령어 집합을 간결하게 설계한 RISC 구조가 탄생하게 되었다[5]. 대표적인 것으로 전술한 바와 같이 Stanford대학의 MIPS 프로젝트와 U.C. Berkeley의 RISC 프로젝트가 있다. RISC 프로젝트와 MIPS 프로젝트는 각각 Sun Microsystems사의 SPARC 프로세서와 MIPS사의 R2000 프로세서로 상용화되어 그후 CPU 설계의 한 축을 이루게 되었다.

이러한 RISC 마이크로프로세서는 간결한 명령어 집합으로 인하여 제어기가 단순화되었으며, 각 명령어의 수행시간을 일정하게 하기 위하여 연산에 필요한 피연산자들을 모두 레지스터에 둘 수 있도록 많은 수의 레지스터 즉 레지스터 파일(register file) 개념을 도입하였다. 이로 인하여 각 명령어 수행을 여러 단계로 나누어 중첩 수행할 수 있는 파이프라이닝이 가능하게 되었다[5, 6]. 이러한 파이프라이닝과 여러 개의 명령어들을 동시에 수행하는 슈퍼스칼라 기법과 같은 명령어 수준 병렬성(instruction-level parallelism)을 이용하는 것이 성능 향상에 많은 도움을 준다는 것을 보임으로써 RISC 구조는 그후 CPU 성능 향상에 많은 기여를 하게 된다.

그러나 CISC 마이크로프로세서에서와 같은 복잡한 일을 수행하는 명령어가 없기 때문에 이를 위해서는 프로그램이 길어져야 하며 따라서 보다 많은 명령어 트래픽과 메모리를 요구하게 된다[4]. 또한 다수의 레지스터로 인하여 이를 잘 사용할 수 있게 하기 위한 기법이 필요하게 되었다. 이러한 문제점들을 해결하기 위해서는 CISC 마이크로프로세서에 비하여 보다 최적화된 컴파일러(compiler)의 도움이 필요하게 되었다.

이러한 CISC와 RISC 구조는 반대 진영의 장점들을 취하는 방향으로 나아가고 있어서 그 경계가 모호해지고 있으며[4], 이는 x86계열의 Pentium에서 슈퍼스칼라 구조를 채택하여 사용하고 있다는 점에서 알 수 있다. 또한 전통적으로 CISC 구조를 사용해 온 인텔이 휴렛패커드(Hewlett-Packard)사와 같이 RISC에 기반한 마이크로프로세서를 개발하겠다고 공표한 데서 알 수 있듯이 점점 RISC 구조 쪽으로 CPU 개발의 추세가 옮겨가고 있다[5].

2.2 슈퍼파이프라인 구조

파이프라이닝(pipelining)은 50년대 후반에 도입되어 주로 대형컴퓨터에서 사용되어져 오다가 80년대 후반에 접어들면서 각 파이프라인 단계(pipeline stage)를 더 작은 n 개의 부단계(substage)로 나누어 수행하는 슈퍼파이프라이닝(superpipelining) 기법으로 발전하여 여러 마이크로프로세서에 채택되기 시작하였다[7,8]. 그림 1은 이러한 파이프라이닝 기법을 보여 준다. (b)에서 $n = 2$ 이며, (a)에서 3개의 명령어를 수행하는

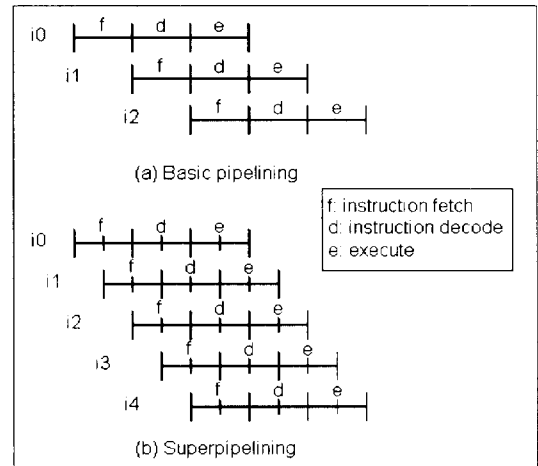


그림 1. 파이프라이닝과 슈퍼파이프라이닝의 명령어 수행 예

동안 (b)에서는 같은 시간에 5개의 명령어를 수행할 수 있어 보다 좋은 성능을 보임을 알 수 있다. 또한 슈퍼파이프라이닝의 각 부단계는 기존의 파이프라인 단계보다 하는 일이 간단하여 더 높은 클럭 주파수에서 동작시킬 수 있어서 더 좋은 성능을 기대할 수 있다 [8]. 이러한 기법을 사용한 마이크로프로세서로는 DEC 21064와 같은 Alpha 칩과 애플-IBM-모토롤라 진영의 PowerPC 계열 등이 있다.

파이프라이닝에서는 명령어들간의 데이터 종속성(data dependency), 자원 충돌(resource conflicts), 분기(branch) 명령어로 인한 절차 종속성(procedural dependency)과 같은 문제로 인한 파이프라인 정지(pipeline stalling)가 발생하며, 이로 인하여 파이프라인 효율(utilization)이 많이 떨어진다 [4, 8, 9]. 이러한 현상은 슈퍼파이프라이닝에서는 더욱 심각한 성능 저하를 가져온다. 이러한 문제점들은 피할 수 없으나 이를 줄이기 위한 방법은 여러 방면에서 연구되고 있다. 분기 명령어로 인한 프로그램 제어 순서의 변화를 예상하기 위한 분기 예상(branch prediction) 방법과 데이터 종속성이나 자원 충돌이 발생하지 않도록 프로그램 순서에 관계없이 서로 충돌이 발생하지 않는 명령어를 우선 실행한 다음 결과 저장시 프로그램 순서에 맞도록 해주는 out-of-order issue에 의한 실행 방법 등이 연구되고 있으며 또한 현재 상용화되고 있는 마이크로프로세서에 채택되어 사용되어지고 있다.

2.3 슈퍼스칼라 구조

슈퍼스칼라 프로세서는 프로그램 내에 존재하는 명령어 수준 병렬성을 좀더 잘 사용할 수 있도록 하는 것을 목적으로 한다. 기존의 스칼라 프로세서가 한 사이클에 하나의 명령어를 수행하는 데 반하여 슈퍼스칼라 프로세서는 $m(m > 1)$ 개의 명령어 파이프라인(instruction pipeline)을 두어 m 개의 명령어를 동시에 수행할 수 있도록 한 것이다. 그림 2는 $m = 2$ 인 슈퍼스칼라 프로세서의 실행 예를 보인 것이다. (a)의

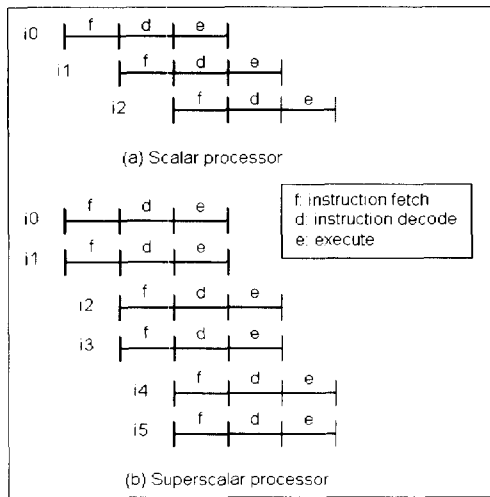


그림 2. 슈퍼스칼라 프로세서의 실행 예

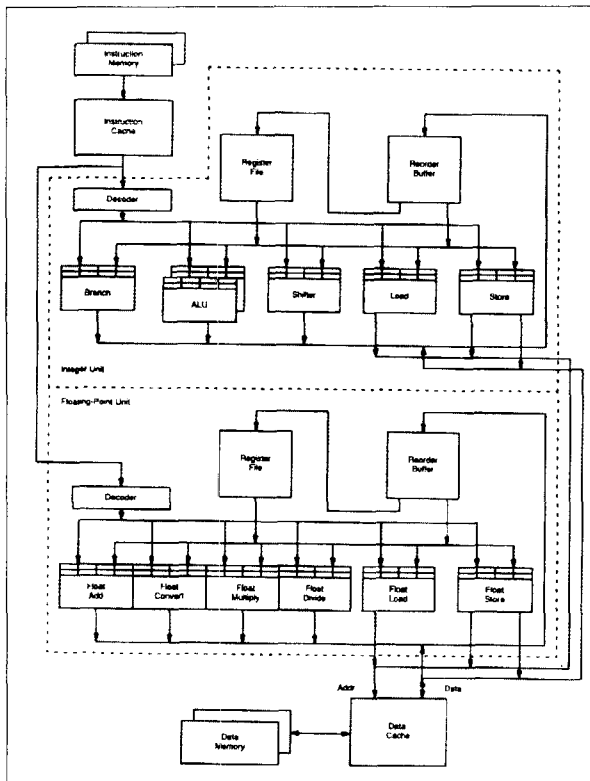


그림 3. 일반적인 슈퍼스칼라 프로세서의 구조

스칼라 프로세서가 3개의 명령어를 수행하는 동안 (b)의 슈퍼스칼라 프로세서는 6개의 명령어를 수행할 수 있어 더 좋은 성능을 보임을 알 수 있다.

현재 대부분의 마이크로프로세서는 파이프라이닝 기법을 기본적인 성능 향상의 수단으로 사용하고 있으며, 슈퍼스칼라 방법과 같이 사용하면 더욱 좋은 성능 향상을 기대할 수 있다. 그러나 데이터 종속성, 절차 종속성, 자원 충돌 등

에 의해 파이프라인이 정지되면 더욱 큰 성능 저하가 발생하게 된다. 따라서 슈퍼스칼라 프로세서에서는 프로그램에 내재된 병렬성을 잘 이용하기 위해서는 슈퍼파이프라이닝 기법에서와 마찬가지로 더욱 최적화된 컴파일러의 도움이 필요하다[4]. 또한 마이크로프로세서 자체에서 성능 향상을 위하여 동적 실행(dynamic execution)과 같은 하드웨어적인 방법을 사용하기도 한다.

일반적인 슈퍼스칼라 프로세서의 구조는 그림 3과 같다 [8]. 정수 연산부(integer unit)에 한 개의 ALU대신 두개의 ALU가 있어 그림 2에서처럼 두개의 연산을 동시에 실행할 수 있다. 이때 필요한 명령어는 명령어 캐쉬(instruction cache)로부터 공급되며 이때 공급되는 명령어의 수를 명령어 이슈율(instruction issue rate)라고 한다. 따라서 동시에 수행할 수 있는 ALU의 수가 많을수록 보다 많이 프로그램에 내재된 병렬성을 이용할 수 있다. 그러나 경험적으로 동시에 실행 가능한 명령어의 수는 대략 3-4개 정도인 것으로 알려져 있으며, 최신의 상용 마이크로프로세서는 보통 2-4 정도의 명령어 이슈율을 가진다. 대표적인 슈퍼스칼라 프로세서로는 DEC 21064, Pentium Pro, PowerPC 601, Sun UltraSparc, MIPS R10000 등이 있다.

3. 산업계의 추세

본 절에서는 현재 상용으로 사용되고 있는 마이크로프로세서중 가장 많은 시장 점유율을 보이고 있는 인텔사의 x86 구조 마이크로프로세서와 가장 클럭 속도가 빠른 DEC의 Alpha 구조 마이크로프로세서에 대하여 기술한다.

3.1 x86 구조

1978년 인텔에서 개발한 8086 마이크로프로세서가 IBM 호환 개인용 컴퓨터에 채택되어 전세계적으로 보급됨에 따라 각광을 받게 되었으며, 반도체 기술이 발전함에 따라서 소프트웨어 호환성을 유지하면서 보다 좋은 성능을 가지는 후속 마이크로프로세서를 개발하였다. 이를 정리하면 표 1과 같다. 전통적으로 인텔의 x86 마이크로프로세서는 서로 길이가 다르며 복잡한 연산을 수행하는 명령어 집합을 가지는 CISC 구조였다. 그러나 Pentium부터는 슈퍼스칼라 기법을 사용하여 RISC 구조의 장점을 채택하기 시작하였으며, Pentium Pro에 이르러서는 *micro-ops*, 슈퍼파이프라이닝 기법 등을 같이 사용하여 보다 RISC 구조에 가까워진다.

8086에서 80486까지는 전형적인 CISC 프로세서의 특징을 가지나 Pentium에 이르러서는 슈퍼스칼라 기법을 도입하여 동시에 2개의 명령어를 수행할 수 있도록 하였다. 또한 새로운 반도체 기술을 적용하여 자체 성능을 더욱 향상시키려는 노력을 하였다[11]. 초기의 P5는 0.8-micron BiCMOS 기술을 사용하여 구현되었으며, 60MHz와 66MHz에서 동작하였다. 이는 Pentium이 마이크로프로세서 시장의 주류가 되기 전에 그 다음 버전인 P54C로 대체되었다. P54C는

표 1. x86 마이크로프로세서의 주요 특징

	8086	80286	80386	80486	Pentium	Pentium Pro
연도	1978	1982	1985	1989	1993	1995
트랜지스터 (millions)	0.029	0.134	0.275	1.2	3.3	5.5 (CPU core)
L1 캐쉬 (data/instr., Kbytes)	X	X	X	X	8/8	8/8
L2 캐쉬 (Kbytes)	X	X	X	X	X	256 또는 512
사이클당 명령어 이슈율	1	1	1	1	2	3
파이프라인 단계	X	X	X	5	5	12-14
데이터폭(bits)	16	16	32	32	64	64
주소폭(bits)	20	24	32	32	32	36

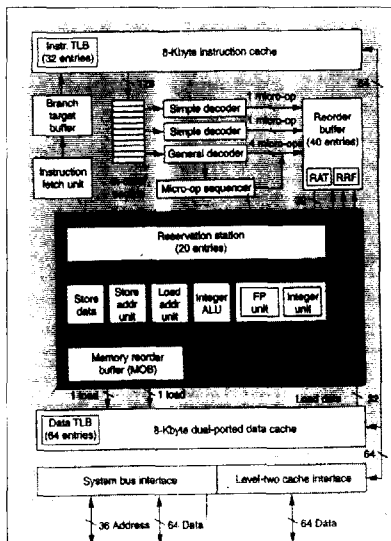


그림 4. Pentium Pro의 구조

0.6-micron BiCMOS 기술로 구현되어 75-120MHz로 동작하였으며, 다시 0.35-micron BiCMOS 기술로 제작되어 200MHz에서도 동작할 수 있는 성능을 가지게 되었다. 이후 인텔에서는 0.28-micron CMOS 기술로 만든 P55C를 내놓으며 기존의 8/8 Kbyte L1 캐쉬를 16/16 Kbyte로 확대하여 윈도우 응용프로그램의 캐쉬 미스율(miss rate)을 감소시켜 성능 향상을 추구하고 있다.

현재 인텔의 최신 프로세서는 Pentium Pro로서, 그림 4와 같은 구조를 가진다[11]. 프로세서 성능이 더욱 고성능화되어 감에 따라서 기존의 복잡한 x86 명령어 집합은 오히려 성능 향상에 저해 요인이 되었다. 그래서 복잡한 명령어 디코더(decoder)를 두어 기존의 복잡한 명령어들을 보다 간단한 명령어의 집합으로 바꾸어 수행하는 방법을 NexGen사에서 도입하여 Nx586에 구현하였다[12]. 이후 AMD사의 K5 프로세서와 인텔의 Pentium Pro에서도 이 방법을 채택

하여 구현되었다. 이런 보다 간단한 RISC형태의 명령어들을 NexGen, AMD, 인텔에서는 각각 RISC86 명령어, R-ops, micro-ops라고 부르며, 기존의 x86 명령어들보다 슈퍼스칼라 구조에서 병렬 수행하기가 쉬워서 성능 향상에 도움이 된다. Pentium Pro는 이러한 micro-ops를 최대 5개까지 동시에 실행할 수 있으며, 밖으로는 3개의 x86 명령어가 실행되는 것처럼 보이는 슈퍼스칼라 구조이다. 또한 기존의 5 단계 파이프라인 단계를 12-14개의 세분화된 파이프라인 단계로 나누어 수행함으로써 보다 높은 성능 향상을 추구하였다. 그리고 프로그램내의 병렬성을 최대한 이용하기 위하여 out-of-order execution을 지원한다. 프로그램 카운터(program counter)뒤에 있는 20-30개의 명령어들을 미리 본 후 서로 충돌이 없는 명령어들을 프로그램 순서에 상관없이 수행함으로써 그 성능을 최대화할 수 있다.

3.2 Alpha 구조

Alpha 구조의 마이크로프로세서는 CPU 구조의 발달과 하드웨어 구현 기술 발전에 따라 탄력성있게 적용할 수 있는 것을 목표로 하여, 클럭 속도, 여러 개의 명령어 이슈, 다중 프로세서 지원에 중점을 두어 설계된 전형적인 64-비트 RISC 구조이다[11, 13]. Alpha 구조는 32-비트 구조의 확장이 아니라 처음부터 64-비트 구조를 채택함으로써 32-비트 구조에서 64-비트 구조로 확장시 발생가능한 문제점들을 처음부터 배제하였다. 또한 다수의 운영체제를 지원하기 위하여 PALcode(Privileged Architecture Library code)라는 소프트웨어 특권층(privileged layer)을 사용하여 특정 운영체제에만 있는 하드웨어 관련 태스크를 기술할 수 있도록 한 점도 다른 마이크로프로세서와 다른 점이다. 1992년에 21064 마이크로프로세서가 개발된 이래 표 2와 같은 특징을 지니는 후속 마이크로프로세서가 개발되었다.

그림 5의 21064 마이크로프로세서는 Alpha 구조를 처음 구현한 것으로서, 0.75-micron CMOS 기술을 사용하여 구현하였으며 1.68 million 트랜지스터의 집적율을 가진다 [13].

표 2. Alpha 마이크로프로세서의 주요 특징

	21064	21064A	21164
연도	1992	1995	1995
클럭속도(MHz)	최대 200	200,233,275,300	366,433,500,600
사이클당 명령어 이슈율	2	2	4
파이프라인 단계 (정수/부동소수)	7/10	7/10	7/9
L1 캐쉬 (data/instr., Kbytes)	8/8	16/16	8/8
L2 캐쉬 (Kbytes)	X	X	96
메모리 데이터 폭 (bits)	64 또는 128	64 또는 128	128

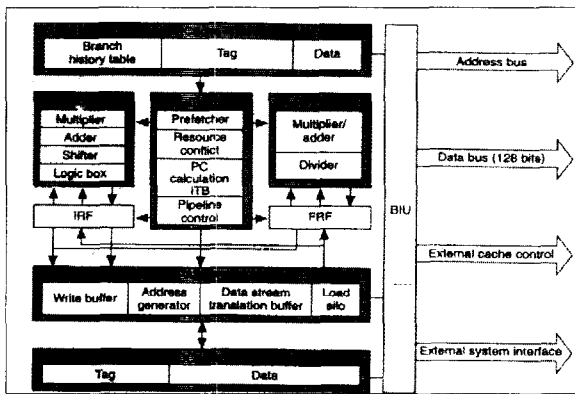


그림 5. DEC 21064 마이크로프로세서의 구조

두개의 명령어를 동시에 실행할 수 있는 슈퍼스칼라 구조이나, 모든 종류의 명령어 쌍에 대하여 실행 가능한 것이 아니라 실행 가능한 명령어 쌍을 제한하였다. 이로 인한 성능 저하는 2% 정도로 작으며, 보다 간단한 구조로 구현할 수 있는 장점이 있다. 또한 21064는 슈퍼파이프라이닝 기법을 사용하는데, 정수 파이프라인과 부동소수 파이프라인이 각각 7단계와 10단계이다. 이 두 파이프라인 안에 최대 22개의 연산(operation)이 각 단계에 들어가 있을 수 있다. 이런 슈퍼파이프라인 구조에서는 분기 명령어가 성능에 많은 영향을 미친다. 21064에서는 conditional move 명령어를 사용하여 분기를 줄이는 방법을 채택하였다. 그러나 피할 수 없는 분기의 경우 후방향(backward) 분기는 선택될 것으로, 전방향(forward) 분기는 선택되지 않을 것으로 예상하는 정적 분기 예상 방법과 2K x 1-비트 분기 히스토리 테이블(history table)을 사용하는 동적 분기 예상 방법을 사용하여 분기로 인한 성능 저하를 줄였다.

4. 맺음 말

1971년 인텔에서 4004 마이크로프로세서를 개발한 이래

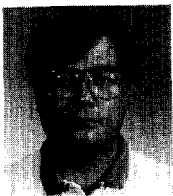
마이크로프로세서 분야는 반도체 기술의 발전과 더불어 그 구조와 성능면에서 괄목할 만한 발전을 하여 왔다. 마이크로프로세서는 워드 프로세싱과 같은 분야에서부터 컴퓨터 시뮬레이션과 같은 고성능을 요구하는 분야에 이르기까지 우리 생활의 많은 부분에서 사용되고 있으며 앞으로도 더욱 그러할 전망이다. 본 고에서는 이러한 마이크로프로세서의 발전 과정에 대하여 고찰하였다. 마이크로프로세서의 클럭 속도는 반도체 기술의 발전과 더불어 2000년경에는 900MHz, 2006년경에는 4GHz에 이를 전망이다 [14]. 1980년대 초의 CISC, RISC 구조에 대한 논쟁은 이제 무의미해졌으며, 양 진영의 장점을 모두 취하는 방향으로 발전하고 있다. 또한 응용 프로그램에 내재된 병렬성을 이용하기 위하여 사용한 슈퍼스칼라 구조는 더욱 각광을 받을 것이며, 이를 위해 보다 최적화된 컴파일러와 라이브러리의 도움이 필요할 것으로 예상된다.

참고 문헌

- [1] P. Wayner, "How Microchips Shook the World," *BYTE*, vol. 21, no. 12, pp.69-74, Dec. 1996.
- [2] J. Markoff, "The Microprocessor's Impact on Society," *Micro*, pp.54-59, Dec. 1996.
- [3] R.G. Daniels, "A Participant's Perspective," *IEEE Micro*, pp.21-31, Dec. 1996.
- [4] K. Hwang, *Advanced Computer Architecture: Parallelism, Scalability, Programmability*, Mc-Graw Hill, Inc., 1993.
- [5] J. Henessy, "RISC Microprocessors," *IEEE Micro*, p.27, Dec. 1996.
- [6] 조정완, 컴퓨터 구조학, 정익사, 1992.
- [7] J.E. Smith and G.S. Sohi, "The Microarchitecture of Superscalar Processors," Univ. of Wisconsin, Aug. 20, 1995.
- [8] M. Johnson, *Superscalar Microprocessor Design*, Prentice-Hall, Inc., 1991.

- [9] J. Hennessy and D.A. Patterson, *Computer Architecture: A Quantative Approach, 2nd ed.*, Morgan Kaufmann Publishers, Inc., 1996.
- [10] Pentium(R) Pro Processor at 150 MHz, 166 MHz, 180 MHz and 200 MHz data sheet
- [11] M. Slater, "The Microprocessor Today," *IEEE Micro*, pp.32-44, Dec. 1996.
- [12] T.R. Halfhill, "Intel's P6," *BYTE*, pp.42-53, April 1996.
- [13] E. McLellan, "The Alpha AXP Architecture and 21064 Processor," *IEEE Micro*, pp.36-47, June 1993.
- [14] A. Yu, "The Future of Microprocessors," *IEEE Micro*, pp.46-53, Dec. 1996.

저 자 소 개



맹승렬 (孟承烈)

1955년 1월 15일생. 1977년 서울대 공대 전자공학과 졸업. 1979년 한국과학기술원 전산학과 졸업(석사). 1984년 한국과학기술원 전산학과 졸업(공학박사). 1988년-89년 펜실바니아대학 교환교수. 1994년-95년 Univ.

of Texas at Austin 교환교수. 1984년-현재 한국과학기술원 전산학과 교수. 관심분야는 parallel computer architecture, dataflow machines, multimedia임.



이종민 (李鍾愷)

1970년 1월 13일생. 1992년 경북대 공대 컴퓨터공학과 졸업. 1994년 한국과학기술원 전산학과 졸업(석사). 1994년-현재 한국과학기술원 전산학과 박사과정. 관심분야는 병렬처리, 워홀라우팅, 멀티미디어 시스템임.