



Alpha CPU 구조

Alpha CPU Architecture

1997년 12월 Microprocessor Report지의 편집자 머릿기사는 "Asian Vendors Take Up Alpha Torch - Mitsubishi, Samsung Show Hope Remains for Alpha in PC Market"라는 제목으로 삼성과 Digital이 차세대 21264 CPU를 0.25um/0.18um으로 스케일링한 PC derivative를 공동개발중이며 생산준비가 거의 완료되었다고 보도하고 있다. Mitsubishi는 아직 0.25um 21164PC도 준비가 되어있지 않으며 Digital 발표에 따르면 Mitsubishi는 특별한 호재가 없는 한 Alpha CPU시장에 적극성을 띄지 않을 것으로 이 기사는 전적으로 삼성의 Alpha CPU에 대한 기대감을 표시한 것으로 삼성의 21264 Alpha CPU의 성능과 발표시기가 세계 CPU 업계의 초점이 되고 있음을 보여주고 있다.



朴 星 培*

Park, Sung Bae

삼성의 21264 Alpha CPU가 세계 CPU 업계의 관심의 초점이 되고 있는 것은 이미 초기 생산 단계에 있는 0.35um 21264만으로 2000년에 발표될 Intel-HP의 차세대 0.18um Merced CPU와 동등의 성능을 갖추고 있으며 0.25um으로 제작될 경우 1.5배, 0.18um으로 제작될 경우 2배 이상의 성능을 갖게됨으로 Intel 진영에서도 Alpha의 성능에 대응할 CPU로 2001년이나 개발될 Merced의 차세대 McKinley 계획을 서둘러 발표하고 있을 정도로 우수한 성능을 갖고 있기 때문이다.

본 논문에서는 20세기 인류 최고의 발명품인 마이크로프로세서의 성능 경쟁에서 삼성 Alpha CPU가 금세기 최상의 성능을 쟁취하며 21세기 마이크로프로세서의 새로운 장을 열어갈 수 있게 된 삼성의 Alpha CPU 구조 기술을 소개하고자 한다.

CPU의 성능은 크게 두 가지 방향으로 상호

보완적으로 발전되어왔다.

곧 단일 클럭 사이클에 실행될 수 있는 명령어의 개수를 최대화하는 것과, 클럭 사이클 타임을 최소화하는 방향이다. 단일 사이클에 실행될 수 있는 명령어의 개수를 최대화할 경우 내부 마이크로아키텍처의 복잡도가 증가하여 사이클 타임이 커지게 되며, 반대로 사이클 타임을 최소화할 경우 주어진 시간 내에 완료할 수 있는 명령어의 개수가 줄어들므로 CPU의 성능은 이 두 가지 상반된 설계 변수를 최적화하며 발전되어왔다.

본 논문에서는 1997년 6월부터 삼성전자에서 개발 판매중인 0.35um 4-Metal CMOS 600MHz 21164 Alpha CPU와 1997년 10월 시제품이 확보된 0.25um 6-Metal CMOS 800MHz 21264 CPU의 아키텍처 소개를 목적으로 한다.

*전자계산기기술사, 삼성전자 System LSI 본부 수석연구원.

1. 서론

마이크로프로세서는 점차 슈퍼컴퓨터의 성능을 단일 반도체 칩에 집적하는 수준으로 개발되어왔다. 실제로 삼성이 1997년 10월, 시제품 개발에 성공한 차세대 21264 CPU의 경우 단일칩에 1,500만개의 트랜지스터를 내장, 800MHz에서 1GHz의 동작 속도를 목표로 초당 최대 5G~6GOPS의 연산이 가능함으로 1990년대의 대표적인 슈퍼컴퓨터인 CRAY-3의 최대 10GOPS에 근접하는 성능을 갖게된다. 마이크로프로세서의 성능은 주어진 일을 얼마나 빠르게 처리할 수 있는지의 표시기준이며 다음과 같이 평가된다.

$$\begin{aligned} \text{Seconds/Program} &= (\text{Instructions/Program}) \\ &\times (\text{Clock Cycles/Instruction}) \\ &\times (\text{Seconds/Clock Cycle}) \end{aligned}$$

Instructions/Program은 CPU의 아키텍처 설계 변수로서 명령어 세트, 컴파일러, OS등에 따라 변화되며 동일한 아키텍처 내에서도 다양한 path length reduction 기법 및 code optimization을 통해 더욱 콤팩트한 명령어수를 얻을 수 있으나 일반적으로 Alpha, SPARC, Power PC, MIPS 등 RISC 계열의 명령어수는 동일한 일을 수행할 때 X86, VAX 등의 CISC 계열의 명령어수에 비해 1.1~1.2배 더 크다.

Clock Cycles/Instruction은 CPU의 마이크로 아키텍처 설계변수로서 파이프라인 하드웨어 구성에 크게 좌우된다. Pentium II의 경우 3-way out-of-order issue, 21164는 4-way in-order issue, 21264는 6-way out-of-order execution 구조를 가지며, Alpha 21364와 Intel-HP의 Merced는 8-way 이상의 동시 실행 파이

프라인 구조를 갖는다. 그러나 파이프라인 효율은 분기 명령어 (branch), cache latency, memory latency, TLB miss, exception, context switching등의 오버헤드 사이클에 크게 좌우되며 6-way 파이프라인이더라도 최소 1개 이하 최대 6개까지의 명령어가 단일 사이클에 처리된다. Seconds/Cycle은 500MHz의 동작 주파수에서 2ns, 1GHz의 경우 1ns로 정해진다.

위 세 가지 변수는 독립적이라기보다는 상호 연관되어 가장 최적화된 값을 가지며 CPU 성능 경쟁의 정점을 이루게 된다.

예를 들어 초기의 Alpha 구조에서 byte 단위의 load/store 명령어를 두지 않음으로 바이트 데이터의 multiplex 시간을 제거하고 (21164 이상의 Alpha CPU는 바이트 단위 load/store를 지원하고 있음), 복수개 명령어의 동시 수행을 가장 크게 제한하는 condition code를 사용하지 않음으로 imprecise interrupt을 허용, 파이프라인 인터록은 물론 예외처리 구현 회로를 단순화 시키며, Memory Barrier를 통해 strict 한 memory ordering 문제를 극복화시킴으로 복수개 프로세서 시스템에서 별도의 지연 없이 초고속 메모리 전송을 가능하게 하는 등의 Alpha 구조 설계 시점부터 이미 Digital의 가장 큰 장점인 초고속 CMOS 회로 설계의 장점을 최대한 활용할 수 있는 구현상의 이점들을 포함시켜 설계되어 있다.

이와 같은 배경을 바탕으로 Intel Pentium II 프로세서와 Alpha CPU간의 성능 비교를 <표 1>에 정리하였다. 주목할 만 한 것은 Pentium II 대비 동일한 0.25um technology를 사용하는 경우 칩 사이즈가 비슷한 0.25um 21264가 6.3배의 월등한 성능을 갖고있음을 알 수 있다.

이 표를 통해 여러 가지를 유추할 수 있는데 예를 들면 명령어 이슈 레이트를 최대로 가정했

〈표 1〉 Alpha-Intel CPU 성능 비교

분 야	Pentium II	21164	0.35um 21264	0.25um 21264
동작주파수	333MHz(1)	600MHz(1.8)	600MHz(1.8)	800MHz(2.4)
SPECint95*	10.3(1)	21.4(2.1)	49(4.8)	65.4(6.3)
발표시기	'97.6	'96.10	('98.8)	('98.10)
Technology	0.25u 4M	0.35u 4M	0.35u 4M	0.25u 6M
Chip Size	203sq.mm	210 sq.mm	320 sq.mm	204 sq.mm
Issue Rate	3	4	6	6
Pipe Stage	12~14	6~12	7~11	7~11
Cache	16K I/16KD	8K I/8K D/96K S	64K I/64K D	64K I/64K D
Transistors	7.5M	9.8M	15M	15M
-Memory	-4M	-6M	-9M	-9M
-Logic	-3.5M	-3.8M	-6M	-6M
Trs./sq.m	37K	47K	47K	74K
Inst./Program	1	1.2	1.2	1.2
M.Cycle/Inst.	0.33	0.25	0.17	0.17
Time/Cycle	3.0n	1.67n	1.67n	1.25n
M.Sec./Prog.	1(1)	0.5(2)	0.34(2.9)	0.25(4)

* Geometric Mean = $\sqrt{\text{SPECint95} \times \text{SPECfp95}}$

** () 발표 예정

을 때 마지막 항목인 Minimum Seconds/ Program의 비율과 실제 성능 비교인 SPECintfp 값의 비율을 비교하면 Pentium II와 21164의 파이프라인 효율이 유사하므로 2~2.1의 비슷한 결과를 얻으며, 0.25um 21264의 경우 Minimum Seconds/Cycle이 Pentium II 대비 4배의 성능을 가지나 실제 성능비는 6.3배에 이므로 잉여의 성능 향상분은 제 6 세대 마이크로아키텍처를 갖고있는 Pentium II 및 21164대비 2배 이상 개선된 제 7 세대 마이크로아키텍처 기술이 기여하고 있음을 알 수 있다.

II. Architecture

Alpha 아키텍처는 1995년에 발행된 Alpha AXP Architecture REFERENCE MANUAL [2]에 규정된 바와 같이 기계어 프로그래머에게 보여주는 Alpha 컴퓨터의 모든 사양으로 정의되

며 명령어 세트, 명령어 형식, OP 코드, 어드레싱 모드, 레지스터와 메모리 등 기계어 프로그래머에 의해 직접 다루어지는 정의들이 포함된다. Architecture(구조)와 Implementation(구현)은 분명하게 구분되어 사용되는데, 21164, 21264등 개별 구현 제품에는 각기 다른 하드웨어 구성, 논리회로 설계, 데이터패스 등을 갖게된다. 매뉴얼은 모두 세 Part로 구성되며 Part I 공통 구조 편에서는 기본 구조 사양, 명령어 형식, 각개 명령어 동작 설명, 시스템 아키텍처와 프로그래밍, 공통 PAL(Privileged Architecture Library) 코드 구조 등에 대해 기술하고있으며 Part II에서는 다시 A, B, C로 나누어 각각 OpenVMS, OSF Unix, Windows NT의 OS 지원을 위한 PAL 구조, 메모리 관리, 프로세서 구조, 예외처리 등을 규정하며 Part III에서는 console 서브 시스템과 시스템 부팅을 기술하고 있다. 상세한 것은 매뉴얼을 참고로 하기 바람이며 여기서는 개략적인 아키텍처를 소개하고자 한다.

1. Alpha AXP 아키텍처 특징

True 64-Bit 구조:

Alpha의 모든 연산은 64비트 레지스터간 64비트 단위에서 수행된다.

초고속 구현 지원:

모든 Alpha 명령어는 32비트로 단순 규칙적 구조를 갖는다.

모든 연산은 레지스터간에서만 이루어지며 특수 목적 레지스터와 컨디션 코드를 배제함으로 복수개의 명령어를 동시에 이슈하는데 파이프라인 인터록을 제거하였다.

이로서 arithmetic trap은 imprecise하게 처리되며 복수개의 명령어가 동시에 trap을 요구할

수 있으므로 파이프라인 구조에서 복수개 명령어의 동시 처리 구조를 상대적으로 쉽게 할 수 있다. 다만 precise한 trap이 요구될 경우에는 trap barrier (TRAPB) 명령어를 사용하여 반드시 해당 명령어의 trap 결과를 보고 다음 명령어를 실행함으로 구현에 유연성을 제공한다. 여기서 precise trap이란 명령어 처리 도중 예외 상황이 발생한 경우 다음 계속되는 명령어들이 절대로 실행되어서는 안된다는 의미이다. 그러나 arithmetic trap의 예를 들면 page fault와 같은 메모리 관리에 해당되는 예외 상황들은 precise하게 처리되어야 한다.

Alpha의 다른 고속 구현 지원 구조로서 jump 명령어에 타겟 어드레스 힌트를 제공함으로 빠르 서브루틴 호출과 복귀가 가능하며 prefetch 힌트를 통해 캐시 히트 비율을 높일 수 있으며, TLB의 granularity hint 비트를 통해 기본 페이지 크기 이상의 대규모 연속 메모리 공간에 대한 TLB의 사용 효율을 높일 수 있다.

■ Multiprocessor의 shared memory 지원:

여러 개의 프로세서가 공유하는 메모리에 읽기/쓰기를 시도할 때 하나의 프로세서가 공유 자원에 액세스 하고 있는 동안 다른 프로세서의 액세스를 저지시킬 필요가 있다.

실행되는 프로세서가 이러한 상태에 있는 것을 critical section이라 부르며, 이때 공유하는 변수를 semaphore라 하는데 프로세서/프로세서간 올바른 데이터 읽기/쓰기를 위해 적절한 synchronization이 필요로 된다. 위에 설명한 바와 같이 하나의 프로세서가 액세스 하는 동안 다른 프로세서에 방해받지 않고 액세스를 완료시키기 위해 하드웨어 혹은 소프트웨어적인 locking 이 필요하다.

Alpha는 하드웨어에서 이와 같은 기능을 지원

하기 위해 primitive operation으로서 load_locked, modify, store_conditional의 동작중 나뉘어 실행 될 수 없는 기본 명령어로서 locking 기능을 제공해주며, 이 과정동안 아무런 인터럽트나 예외, 다른 프로세서로부터의 쓰기 간섭이 일어나지 않으면 conditional store가 성공함으로서 critical section에서 벗어나게 된다. 만약 이 과정중 방해를 받게되면 프로그램은 원점으로 되돌아와 이 과정을 반복하게 된다.

Alpha 아키텍처는 MP 시스템에서 어떤 종류의 엄격한 (strict) 읽기/쓰기 순서화도 규정하지 않는다. 엄격한 읽기/쓰기란 하나의 프로세서에 의해 발생된 읽기/쓰기는 반드시 모든 프로세서에 발생한 순서대로 제공되어야 함을 의미한다.

Alpha 아키텍처는 하나의 프로세서에서 A에 쓰고 B에 쓰는 작업시, A에 쓰는 것이 실패하고 B는 성공적으로 쓰이고 나중에 A에 쓰기 재시도를 하여 성공하면 다른 프로세서들은 순서에 관계없이 B를 먼저보고 나중에 A를 보더라도 프로그램 실행에 지장이 없어야하며 이러한 유연성을 허용하지 않고 엄격한 모델을 채택했던 VAX 구조는 주기억장치에 파이프라인화 된 쓰기, 여러 개의 뱅크를 갖는 캐시, 네트워크 라우팅, 크로스바 메모리 스위치 등을 구현하지 못함으로 많은 성능상 제약이 따랐었다. Alpha 아키텍처에서 엄격한 메모리 모델이 필요할 때는 Memory Barrier (MB) 명령어를 사용함으로 강제적인 순서화 구조를 제공한다.

[참고] SPARC V9은 RMO(Relaxed Memory Order)를 새로이 도입함으로써 프로그램 카운터상 순서만 올바르게 메모리 액세스상 load/store는 어떤 순서대로 진행되어도 좋도록 성능을 개선하였으나 V8 호환성 보장을 위해

PSO(Partial Store Order)/TSO(Total Store Order) 메모리 모델도 병행 규정하고있는데 비해 Alpha는 초기부터 RMO를 채택, MP에서의 아키텍처적인 성능 개선이 고려되었다.

Flexible Privileged S/W Library 지원:

대부분의 시스템 설계시, 사용자와 OS간에 서로 일관성을 갖고 공유할 수 있는 기능들이 필요하다. 이러한 공통 기능 interface가 제정되면 서로 다른 시스템 개발시에도 기본 기능 라이브러리는 아무런 수정 없이 그대로 사용할 수 있는데 Alpha는 PAL(Privileged Architecture Library) code를 제공함으로써 구현의 유연성을 최대화 시켰다.

PAL code 에는 TB miss fill, interrupt acknowledge, vector dispatch 등의 low-level hardware를 직접 제어하는 기본적인 atomic한 과정이면서 긴 명령어 순서에 따라 실행되는 다음과 같은 기능들이 포함된다. VAX에서는 주로 microcode로 제공되던 기능들에 해당된다.

- ▶ 복잡하면서도 전체적으로 atomic한 동작을 요구하는 명령어 그룹
- ▶ Memory 관리, TB 관리
- ▶ Context Swapping
- ▶ Interrupt & Exception Dispatching
- ▶ Power Up 초기화 및 booting
- ▶ Console 기능(시스템 소프트웨어 boot-strap 등)
- ▶ Instruction Emulation without H/W

예를 들어 32엔트리의 TB를 갖는 시스템과 1024엔트리를 갖는 시스템에서 각개의 TB 미스 PAL 코드를 가짐으로서 OS는 특정 시스템의 TB 엔트리 숫자를 알 필요가 없어진다.

따라서 특정 OS에 따라 특정의 PAL 코드들

이 대체되어 포팅이 이루어짐으로 PAL 코드들은 복수개의 OS와 플랫폼의 easy porting을 위해 효과적이다. PAL 코드 환경에서는 머신의 모든 상태가 제어될 수 있어야 하며 인터럽트가 금지됨으로서 전체 명령어 실행 순서가 atomic하게 수행됨으로, low level 시스템 하드웨어 액세스를 가능케 하고, I-stream의 메모리 관련 trap을 금지시킴으로 TB fill과 같은 기능을 가능케 한다.

2. Data Format

컴퓨터의 데이터 타입은 바이트 데이터 구성 순서에 따라 Little-Endian 방식과 Big-Endian 방식의 두 가지로 나뉘어진다.

Alpha와 X86은 모두 Little-Endian을 기본으로 하는데 이것은 스타트 어드레스가 바로 바이트 0에 해당되며 메모리 어드레스가 1 증가 될 때마다 바이트 어드레스가 증가하는 형태로 달걀로 보면 뾰족한 면이 아래이고 위로 갈수록 증가하는데서 이름이 기인하며 SPARC 등은 Big-Endian 타입으로 스타트 어드레스가 가장 큰 바이트 데이터를 갖는다. 곧 64비트 8바이트 데이터의 경우 Big-Endian은 바이트 0,1,2,3,4,5,6,7로 메모리 어드레스 0에 액세스하면 바이트 7이 읽혀오며 어드레스 1에 액세스하면 바이트 6 등이 순차적으로 읽혀지게 된다.

Alpha에서는 부트 시간동안 endian 타입을 선정하도록 허용하며 구현시 실행 시간동안 다이내믹하게 변화될 기능을 반드시 요구하지는 않는다. 명령어는 모드에 관계없이 언제나 Little-Endian으로 인식되어야 하며 컴파일러, 링커, 디버거들에게 모드 정보가 미리 제공되어야 한다. 만일 Big-Endian으로 선택되면 long word 액세스는 모두 Virtual Address Bit 2를 반전시켜

사용해야 하며 이로서 quadword 바운더리로 된 데이터 액세스시 올바른 long word 데이터를 갖고 올 수 있다. 또한 바이트 연산시 Rb<2:0>을 반전시킴으로서 예를 들어 바이트 5를 추출할 것을 바이트 2를 추출하는 효과를 가져오게 한다.

- 32개의 64비트 Integer 레지스터와 32개의 64비트 Floting Point 레지스터를 갖는다.
- 4개의 정수 데이터 타입을 지원한다.
 - 8비트 byte
 - 16비트 word
 - 32비트 long word
 - 64비트 quad word
- 5개의 부동 소수점 데이터 타입을 지원한다.
 - IEEE S_Floating [32-Bit]
 - IEEE T_Floating [64-Bit]
 - IEEE X_Floating [128-Bit]
 - VAX F_Floating [32-Bit]
 - VAX G_Floating [64-Bit]
 - * Long Word integer in FPR
 - * Quad Word integer in FPR

3. Alpha AXP 레지스터

- 64비트 Program Counter<63:2> <1:0>는 RAZ/IGN

[참고] 레지스터 내용에 대해 다음과 같은 naming convention을 사용함.

- * MBZ: Must Be Zero
- RAZ: Read As Zero
- SBZ: Should Be Zero
- IGN: Ignore

IMP: Implemenattion Dependent

RAO: Read As One

RC: Read to Clear [written by H/W]

RO: Read Only [written by H/W]

RW: Read Write

* Undefined: Privileged S/W만 생성

프로세서를 halt 하거나 정보를 잃을 수 있음.

* Unpredictable: 프로세서의 기본 오퍼레이션에 영향을 주지 않음. 어느 S/W든 생성 가능

※ 64-Bit X 32 [R0-R31] Integer Register:

다른 레지스터와 달리 R31은 읽을 경우 0, 쓸 경우 unpredictable 연산을 하게된다. 따라서 R31을 destination으로 하는 명령어가 예외를 발생시키면 결과는 unpredictable이며 어떤 경우라도 이런 명령어의 fetch 과정에서 발생하는 예외도 항상 알려지게 된다.

STx_C R31, disp(Rb)와 같은 경우 공유 장소에 0를 쓰며 lock_flag를 reset 시키는 좋은 방법으로 생각되나 결과는 lock_flag와 타겟 어드레스의 값이 unpredictable로 처리된다. LDx_L R31, disp(Rb) 도 마찬가지로 lock_flag 와 locked_physical_address를 unpredictable로 처리한다.

BR, BSR, JMP, JSR, RET, JSR__COR-OUTINE에서 R31을 Ra로 지정하면 분기 명령어는 예정대로 실행되고 복귀 어드레스는 R31에 당연히 저장되지 않는다.

※ 64-Bit X 32 [F0-F31] Floting Register : F31은 R31과 마찬가지로 true-0로 처리되는 데 sign, exponent, fraction 모두 0로 간주된다.

※ Lock Register: MP synchronization을 위해 lock_flag와 locked_physical_address 의

2개 locked register를 제공한다.

LDx_L이 아무 에러 없이 종료되면 프로세서는 프로세서당 하나씩 있는 locked_physical_address register에 타겟 어드레스를 저장하며 마찬가지로 프로세서당 하나씩 할당되는 lock_flag를 세트한다. 다음에 나오는 STx_C 명령어가 실행될 때까지 lock_flag가 세트 되어있으면 STx_C를 실행하되 그렇지 않으면 store는 실행되지 않는다. 예를 들어 A 프로세서의 lock_flag가 세트되어있는데 B 프로세서가 A 프로세서의 locked physical address range에 store하면 A의 lock flag는 리셋되어 A 프로세서가 lock 한 LDx_L의 결과를 무효화시킴으로써 다시 A 프로세서로 하여금 lock range를 액세스하도록한다. 여기서 lock_range는 페이지 사이즈 이내 16바이트 이상의 2**N으로 N은 구현에 따라 달라진다. lock_flag는 또한 CALL_PAL REI/rti/rfe 명령어 등을 만나면 clear되며 기타 CALL_PAL 명령어는 lock_flag에 unpredictable 결과를 낸다. BR, BSR 및 Jumps 등과 같은 taken branch 경우 프로세서의 lock_flag는 unpredictable 결과를 낸다. 조건 분기 명령어도 분기가 이루어질 경우는 lock_flag를 clear 하지 않는다.

● PCC(Processor Cycle Counter) Register

Low 32-Bit은 unsigned wrapping counter이며 High 32-Bit은 OS에 따라 달라지는데 PCC_CNT는 ns 단위의 시간 측정에도 유효하게 사용된다.

● Optional Registers

Memory Prefetch Register : FETCH 명령어 실행시 상태를 기록하는 두개의 레지스터 세트

VAX Compatibility Register:

RC-Read and Clear intr_flag

RS-Read and Set intr_flag

RS와 RC 명령어 사이에 예외처리 없이 진행되었는지 여부를 판별하는데 사용한다. intr_flag는 CALL_PAL REI 명령어를 만날 경우 clear 된다.

III. 결론

CPU는 첨단 전자 산업의 핵심 부품이자, 정보통신 산업의 최첨단에 위치한 전략적 제품으로서 각종 소프트웨어에서 하드웨어에 이르기까지 전자, 통신, 정보 산업 전 분야 제품군의 시스템 동작 특성과 성능, 규격을 결정짓는 핵심 기술중의 핵심 부문에 해당된다. 또한 CPU 기술은 초고집적 초고속 반도체 기술의 최첨단 제품군으로서 여기서 파생되는 기술은 반도체 산업 전분야를 선도하게 된다.

1GHz Full Custom Dynamic CMOS 설계, deep submicron 공정, 초고속/대규모 핀/대열량 패키징, 초고속 시험 기술, 초대규모 초고속 집적회로 설계 검증용 CAD, 시스템 verification 등의 기술력은 곧 최상의 경쟁력을 갖는 반도체 제품군 개발을 선도하게 됨으로 최고 성능의 CPU 개발은 국내 산업 전반에 걸쳐 핵심 역할을 담당하게 된다.

결론적으로 Alpha CPU 구조 기술을 소개하며 향후 CPU 산업의 발전을 통해 얻을 수 있는 국내 산업 발전의 부가효과를 정리하는 것으로 본 논문을 맺고자한다.

1. 한국산 제품의 Brand Name 시그

최근까지도 국내 산업 전 분야 제품은 저가 제

품의 이미지를 갖고 있으며 세계 최상위 제품에 대한 인지도에서는 삼성 반도체가 메모리 제품에서 세계 1위의 위치를 수년간 지켜오며 반도체 분야의 독자적인 이미지 구축을 해왔으나, 여기에 더해 비메모리 반도체, 첨단 반도체중에서도 최첨단의 CPU, 그것도 금세기 최고의 성능을 갖는 CPU의 개발, 공급을 통해 타 비메모리 제품에 대한 네임 브랜드의 고급화는 물론 한국산 제품들에 대한 이미지 제고를 통해 고부가가치 효과를 충분히 이용할 수 있도록 전략적인 차원에서 활용이 가능하다.

실제로 영화 타이타닉의 그래픽 효과를 처리하는데 한국산 Alpha CPU가 사용된 내용을 홍보할 경우 국내 기술력에 대한 세계인들의 인지도를 높일 수 있으며 CPU 자체 가격이 수천달러대에 이름으로 초고가 고성능 제품에서의 한국 제품의 경쟁 가능성을 부각시키며 기타 산업 분야에서도 세계 1등 제품군의 확산에 기여할 수 있다.

2. 반도체 제분야 경쟁력의 획기적 제고

Alpha CPU는 내부에 메모리 분야의 Cache SRAM을 비롯해 비메모리 분야의 초고속 회로를 내장하는데, CPU의 회로 기술은 초기의 Full Custom에서 Cell 및 ASIC 기술을 활용한 Semi Custom으로 발전되어왔으나, Alpha의 경우 철저한 Full Custom 방식을 최적화 시켜, 단위 면적당 트랜지스터 내장 규모에서 경쟁 제품의 2배 이상, 동작 속도 2배 이상 등, 성능-집적도 효율 면에서 4~5배의 Full Custom 설계 기술력과 고속 트랜지스터 및 interconnection 공정 기술을 사용함으로 최고 성능을 달성해 왔다.

반도체 제분야에서 이와 같은 최고성능의 CPU에 사용된 기술들을 효율적으로 활용할 경

우 메모리 및 비메모리 제품 경쟁력이 획기적으로 개선 될 수 있으며 특히 이러한 성능이 가능 한데는 Digital 고유의 CAD 기술에 설계 및 공정의 많은 know-how들이 expert tool로서 집적됨으로 설계자의 독자적인 판단에 따른 설계 오류를 최대한 제거하는데 이러한 CAD tool의 도입, 활용, 개선을 통해 21세기 새로운 반도체 제품 경쟁력 확보가 가능해지는 원천 기술 확보가 가능하다.

3. 국내 전자산업의 경쟁력 확보를 위한 방안으로서의 시사점

국내 전자산업은 monitor, keyboard, HDD, CD-ROM, network, cellular, 통신, 인터넷, 콘텐츠 등 광범위한 시스템들의 Vertical Integration 시스템 산업 기반을 잘 갖추고 있음으로 이 산업 분야 전체를 연결시켜주는 시너지 효과를 통해 제대로 결합시킬 경우 세계 최정상 의 이상적인 제품 경쟁력을 갖추게 된다.

이중에서 현재까지 시스템의 핵심을 담당하며 성능, 소프트웨어, 가격 등 모든면에서 중앙에 위치하여 시스템 산업을 주도해온 마이크로프로세서의 국내 고유 기술화가 가능하게됨으로 Alpha CPU 기술을 바탕으로 Vertical System Integration의 모든 component중 항상 부족해 온 마이크로프로세서에 국내 독자의 solution을 갖추게 되어 종속적으로 항상 외국의 뒷북을 쳐온 국내 전자 산업에 creative한 신규 시장 창출을 앞서 제공할 수 있어 능동적으로 세계 전자 산업을 리드 해 갈 수 있는 기반을 구축한다.

여기에 마이크로프로세서를 중심으로 확산되는 관련 기술, Horizontal Technology의 발전 - 시스템, 보드, OS, 컴파일러, 그래픽 S/W, 콘텐츠 등의 멀티미디어 사업 등을 유도하게되어 세계

전자 산업에서 한국의 현재 위치를 수동형, 저가형에서 창의형, 고부가가치형 사업으로의 Vertical -Horizontal 국가 전략 산업화를 가능하게 하는 핵심 기술로서의 가치를 갖는다.

4. 고부가가치 CPU 산업 육성

1997년 GE가 1400억달러의 매출에 40억달러의 순이익을 내는 동안 CPU의 대표적 회사인 Intel은 200억달러의 매출에 50억달러의 순이익을 내는 가장 이상적인 고부가가치 사업을 일구어 냈다.

Alpha CPU를 이와 같은 고수익 CPU 사업에 진출하기 위한 기반 기술로서 압도적인 성능차이를 갖는 기술적 우위를 바탕으로 세계 CPU 산업계에 변화기를 이룬, 고부가가치 고수익 반도체

제품 확보가 가능하다.

5. 인력 및 기술 축적에 의한 반도체 종합 서비스 산업 기반 구축

21세기의 반도체 경쟁력은 동작 속도와 집적도로서, 메모리가 1G DRAM으로 대표된다면 CPU는 1GHz의 동작 속도로 대표되며 1억개 상당의 트랜지스터를 단일칩에 내장하여 최상의 성능을 얻는 경쟁이 되며 현재 개발되고 있는 CPU 기술을 제대로 소화하고 도착화시킬 경우 ASIC으로 대표되는 시스템의 반도체화 서비스 산업에서 최고 수준의 설계 기술과 공정 기술로서 최상급 반도체 설계 및 ASIC, 제품 foundry, 개발 공급 등 고성능 고수익 종합 반도체 서비스 산업의 국내 기반 구축이 가능하다.

(원고 접수일 1998. 7. 15)

참고문헌

1. William Grundamnn et al., "Designing High Performance CMOS Microprocessors Using Full Custom Techniques," in DAC, 1997, pp.722~727
2. R. Sites, Ed., Alpha AXP Architecture REFERENCE MANUAL. Newton, MA, Digital, 1995