

## 중간 리포트

### 0. 요약

반세기의 컴퓨터의 설계는 많은 것들을 바꿔놓았고 많은 법칙들을 깨뜨렸다. 컴퓨터의 설계의 시작이라고 말할 수 있는 IBM system을 소개하면서 글은 흘러간다. IBM은 모든 기계에 대해 하나의 바이너리 호환 명령어세트인 360 system을 개발했다. 이러한 개발에서는 설계하기 어려운 제어 로직에 마이크로 프로그래밍을 접목시킨다. 이러한 배경속에서 메인 프레임 설계는 꽃을 피우게 된다. 이러한 모습들은 프로그래머들이 작업을 더 쉽게 할 수 있게 만들어 주었으며, CISC의 탄생을 의미 한다. 발전 속에서는 다른 가능성이 나오게 된다. 기존에 사용되던 마이크로프로세서는 너무 느렸다. 이러한 문제를 해결하기 위해서 성능저하의 원인중 하나였던 제어 쪽에서 문제를 단순화 하기 위해 명령어 수를 최소한으로 줄이기로 하였다. 해결책으로 파이프라인데이터 경로와 빠르고 마이크로 프로그래밍되지 않는 제어 회로를 사용해서 근본적으로 간소화된 프로세서 아키텍처를 개발했는데, 이 아키텍처는 RISC를 의미한다. CISC보단 빨랐지만 CISC에 비해서 프로그래머 친화적이지 않기 때문에 최적화된 컴파일러의 개발의 필요를 의미하게 된다. 이러한 성과를 보이는 중 더 나은 ISA를 개발하기위해서 노력했지만 이것은 실패로 이어진다. 특히 VLIW 같은 프로세서는 VLIW 소프트웨어 컴파일러가 고급 소스 코드에서 생성 된 명령을 VLIW 기계코드로 스케줄하고 압축하는 방법을 기대했지만 여러가지 문제들로 인해서 실패하게 된다. 그렇게 기술의 발전은 고도화되며 무어의 법칙 암달의 법칙등 여러가지들의 법칙들을 성공적이게 보이게 했지만 최근에 와서는 이러한 법칙들이 죽었음을 그래프로 보여준다. 프로세서의 성능향상이 연간 3%로 급격히 떨어졌기 때문이다. 하지만 우리는 가능성을 과거의 실패에서 볼수있다. 과거에는 VLIW가 실패했지만 특정용도로 최적화 시킨다면 성능이 좋아지는 것을 확인했기때문이다. 이러한 앞서 말한 법칙들은 죽었다고 사회가 공공연하게 받아들여지지만 그 죽음 가운데 피어난 꽃을 가능성으로 기대해도된다.

1. Intel 사의 Itanium, EPIC과 같은 VLIW processors들이 시장에서 성공적이지 않았던 이유는?

1) 분기는 예측할 수 없으므로 병렬작업을 예약하고 VLIW명령어로 패킹하는 작업이 복잡해진다.

2) 예측할 수 없는 캐시는 느린 실행을 놓치고 다양한 실행 대기 시간을 만들게 된다.

3) VLIW ISA는 코드의 사이즈를 증가시킨다.

4) VLIW 머신에 적합한 최적화 컴파일러는 만들기가 너무 어렵다.

2. Domain Specific Architectures 가 앞으로의 프로세서의 주요 발전방향으로 주목받는 이유는 무엇인가?

VLIW 아키텍처는 범용 컴퓨팅을 위한 시작에 실패했지만 분기가 훨씬 적은 DSP 애플리케이션에는 적합하기 때문이다. 예를 들어서 실리콘 한 조각에 65,536 개의 MAC(multiply / accumulate)단위의 대규모 배열을 사용하여 DNN실행을 가속화 하는 Google의 TPU는 기존 서버 CPU보다 약 30배, GPU보다는 15배 빠르며 전력 또한 낮아서 성능/ 전력 이점이 각각 60배와 30배로 나타나진다.

3. 사용자의 요구는 높아지지만 프로세서의 속도는 더 이상 빨라지기 힘든 기술적 장벽을 극복하기 위한 방법은 어떤 것이 있는가?

공유메모리 시스템의 형태로 멀티프로세서가 있다 CMP 나 NUMA같은 프로세서로 CMP의 경우 Chip Multiprocessor로 하나의 칩에 여러개의 코어를 장착한 형태의 프로세서이며 NUMA의 경우 각프로세서에 독립적인 메모리를 할당하여 각 프로세서는 자신에게 할당된 지역 메모리로 빠르게 접근하며 원격 메모리로는 느리게 접근되게 하는 형태이다. 그 밖에는 분산 메모리 시스템으로 노드라고 부르는 독립적인 컴퓨터를 고속의 네트워크로 연결한 형태로 클러스터모드라고 생각하면된다.