

## 논리형 추론컴퓨터 (Logic Machine)

趙 廷 完

(正 會 員)

韓國科學技術院 電算科 教授

## I. 서 론

다음 세대의 컴퓨터는 인공지능을 가진 지능형 컴퓨터가 주축을 이루리라는 것은 이미 널리 인정되고 있다. 이런 인공지능형 컴퓨터의 목적은 정보와 지식을 효과적으로 수집하여 표현하고, 이들을 처리함으로써 여러 응용분야에 활용할 수 있도록 하는 것이라 할 수 있다. 따라서 인공지능형 컴퓨터에서는 지금까지 사용되어 왔던 수치계산용 컴퓨터와는 달리 비교, 선택, sort, pattern matching과 논리적 집합 연산과 같은 기호처리(symbolic processing), 주어진 지식 표현방법을 이용하여 문제를 풀어갈 때 발생하는 탐색 위주의 작업등의 특수한 처리 방식들이 주로 사용되며, non-deterministic한 알고리즘이 많이 사용된다는 특징이 있다.<sup>[1]</sup> 이러한 지능형 컴퓨터 시스템은 정보의 수집을 위해서 기호, 영상, 문서, 음성 등을 직접 입력할 수 있는 man-machine interface와 입력된 정보를 효과적으로 처리할 수 있도록 지식을 표현하는 지식 베이스, 그리고 지식베이스에 표현된 지식을 바탕으로 주어진 문제를 풀어나가는 추론 기관으로 나눌 수 있는,<sup>[2]</sup> 추론 기관을 효과적으로 실현하기 위해서는 기존 von-Neumann형 구조보다도 병렬수행과 분산처리 능력을 가진 새로운 형태의 컴퓨터 구조 설계방식이 요구되는 실정이다.<sup>[3,4]</sup>

인공지능형 컴퓨터에서 요구하고 있는 처리 방식의 특징들을 기존의 컴퓨터처리 방식과 비교해보면 표 1 과 같이 나타낼 수 있다.

현재 개발 되었거나 개발중인 지능형 컴퓨터의 구조들을 크게 세가지로 분류할 수 있는데, 그림 1 과 같이 언어중심의 기계, 지식표현 중심의 기계, 그리고 지능형 인터페이스 기계등으로 나눌 수 있다. 언어중심의 기계와 지식표현 중심의 기계들은 추론기관을 설계하는 기준을 달리하면서 생겨난 설계방식으로 생각할 수

표 1. 지능형 프로그램 방식의 특징

특 정	AI Programming	Conventional Programming
처리방식	기호처리	계산중심
처리기술	Heuristic Search	Algorithmic
답을 구하는 과정에 대한 정의	불 분명	정확한 기술
구하려는 답의 성격	만족스러운 해답	최적의 해답
제어 / 데이터 구분	분 리	혼 합
Knowledge	부 정 확	정 확
변경의 빈도수	많 음	적 음

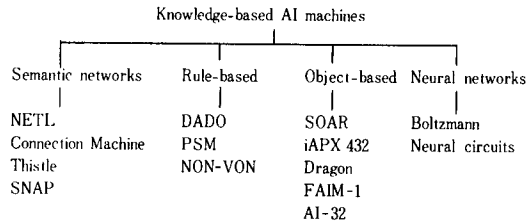
있다. 언어중심의 기계는 지능형 컴퓨터에서 요구되는 지식표현 방식을 일반적인 관점에서 지원하려는 것이며 지식표현 중심의 기계는 특정 지식표현 방식에만 초점을 맞춰 설계한 것으로 언어중심의 기계보다는 좀더 특수한 설계방식들을 사용한다.<sup>[1]</sup> 지능형 인터페이스 기계는 man-machine 인터페이스의 기계적 구현이라 생각 할 수 있다. 위의 세가지 기계들을 현재 알려진 여러 상용 혹은 연구용 기계들과 함께 자세히 분류해보면 그림 1 과 같다. 이들 각각이 추구하고 있는 구조 설계 방식들은 서로 다른 특징들을 나타내고 있으므로, 이 글에서 기술하고자 하는 추론컴퓨터는 resolution을 논증방식으로 사용하는 논리 프로그램을 위해 데이터를 추출하고, pattern matching과 unification을 효과적으로 수행하도록 설계된 컴퓨터를 지칭하는 것으로 한다.

이 글에서는 추론 컴퓨터의 형태를 수행방식에 따라 순차 수행방식과 병렬 수행방식으로 크게 나누어 요약하고 그 특징들을 비교 분석하기로 한다.

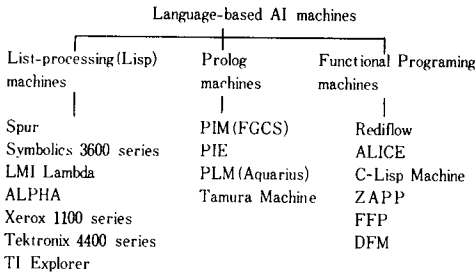
## II. 순차수행 추론컴퓨터

## (Sequential Inference Machine)

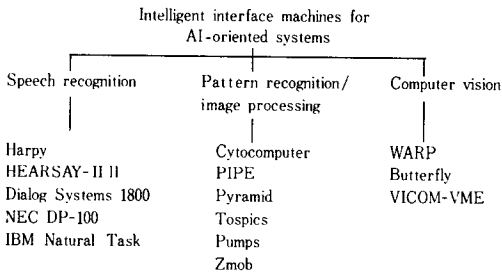
순차수행 추론컴퓨터의 개발은 Lisp maching과 마



(a) 지식 표현 중심의 기계들



(b) 언어 중심의 기계들



(c) 지능형 인터페이스 기계들

그림 1. 지능형 컴퓨터의 세가지 유형

찬가지로, prolog 프로그램 개발 환경을 위한 전용 컴퓨터의 실현이라는 실용적인 요구와 새로운 구조를 가진 컴퓨터 시스템을 개발하기 위한 연구적인 측면에서 추진되어 왔다. 순차수행 추론컴퓨터는 크게 2 가지로 나눌 수 있는데 첫번째 논리언어인 prolog를 중간코드로 바꾼 후 마이크로 프로그램에 의해 interpreting 하는 방식이며, 두번째는 논리 언어를 기계어 코드로 완전히 변환하는 컴파일링 방식이다. 전자에 해당되는 것으로 ICOT의 PSI-I, PSI-II, Kobe 대학의 PEK 등이 있으며, Warren과 Tick이 제안한 pipelined prolog machine, Berkeley의 PLM(programmed logic machine), NEC의 CHI(co-operative high performance sequential inference machine) 등이 후자에 해당된다고 할 수 있다.

1. ICOT의 PSI-I과 PSI-II

PSI-I 은 제 5세대 컴퓨터 시스템 연구개발 프로젝트의 일환으로 ICOT가 개발한 horizontal micro-program 방식으로 구성된 최초의 순차수행 컴퓨터로 그림 2 과 같이 구성되어 있다.<sup>(4,5)</sup> 수행 속도는 약 30K LISP (LIPS : logical instructions per second)이며 최대 기억 용량은 16M word로 되어 있다.

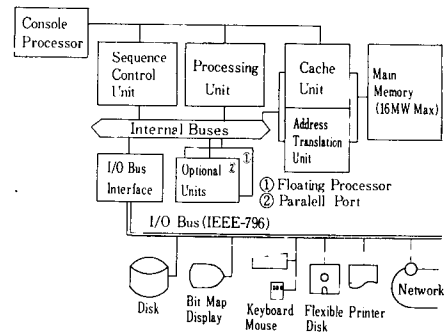


그림 2. PSI의 구성

H/W의 설계는 TTL로 되어 있으며, 운영체제는 prolog에 object-oriented 기능을 부가한 시스템 서술 언어인 ESP(extended self-contained prolog)로 코딩된 SIMPOS이다. ESP 프로그램은 컴파일러에 의해 PSI의 기계어에 해당하는 KL0로 번역된다. KL0는 prolog에 몇가지 확장기능을 첨가한 언어로 마이크로 프로그램에 의해 interpreting 방식으로 직접 수행된다. PSI-I 은 이미 제작이 완성되어 사용하고 있으며, 1987년 전반기에 발표할 예정으로 PSI-II가 개발되고 있다. PSI-I 과 PSI-II 의 기능 비교표는 다음과 같다.<sup>(6)</sup>

2. Kobe 대학의 PEK

PEK은 prolog 프로그램을 고속 수행하기 위해 개발

표 2. PSI-I 과 PSI-II 의 기능 비교

	PSI	PSI-II
Size	Big refrigerator	Desk side
Device	TTL(fast)	CMOS Gate Array LSI
Cycle time	200 nsec.	200 nsec.
Cache memory	4 KW×2	4 KW×1
Main memory	16MW (Max.)	64MW (Max.)
Max. num. of processes	64	No limit
Machine code type	Table type	Instruction type
Structure rep.	Sharing	Copying
Performance (ave.)	30 KLIPS	Around 100 KLIPS

된 실험용 컴퓨터로서 PSI와 마찬가지로 마이크로 프로그램에 의한 해석 방식을 사용하고 있는데, sequencer 및 ALU는 비트 슬라이스 LSI(Am 2903A, Am 2909A)를 사용하고 있다. PEK은 MC 68000(CP/M 68K)을 주 컴퓨터로 사용하는 back-end 프로세서의 형태를 취하고 있으며 unification과 backtraking 작업을 위한 하드웨어 stack들을 갖추고 있다. 그림3은PEK의 하드웨어 구성도를 나타내고 있다. PEK의 내부 버스 폭은 34bit이며, 3개의 버스로 구성된다. 마이크로 명령어의 사이클 시간은 가변적이지만 평균 180nsec 정도이다. 1word는 96bit로 구성되어 있고 16K word의 제어 기억장치를 가지고 있으며 pipeline 레지스터를 사용하여 명령어를 prefetch하여 사용한다.

PEK의 성능에 대한 전반적 평가는 알 수 없으나 대략 40KLIPS의 성능을 나타내는 것으로 알려져 있다.<sup>17)</sup>

### 3. Warren과 Tick의 Pipelined Prolog Machine

WAM(Warren의 abstract machine)을 실제 구현하기 위해 현재 구성되고 있는 컴퓨터로 LISP machine과 대항할 수 있는 정도의 성능을 목표로 하고 있다. Proloy 프로그램은 컴파일러에 의해 Warren이 정의한 structure copying 방식의 기계어 명령어로 변환

되어 수행되며 pipeline식 제어를 사용하고 있다. 이 prolog machine의 중추는 기억장치 시스템, 명령어 장치(I-unit), 수행장치(E-unit)로 이루어지는데 기억장치 시스템은 기억장치 액세스를 중첩(overlap) 처리 기법을 사용하여 인터리빙함으로써 기억장치 병목현상을 줄이도록 설계되어 있다. I-unit는 E-unit에 명령어를 공급하는 작업을 수행하며, E-unit는 3개의 pipeline 단계로 구성되어 수행 속도를 가속시킬 수 있도록 설계되어 있다. Pipelined prolog machine의 성능은 약 450K LIPS이며, Symbolics 3600 위에 firmware로 구현했을 때는 110K LIPS로 추정되고 있다.<sup>18)</sup>

### 4. NEC의 CHI(co-operative high performance sequential inference machine)

일본의 제 5세대 컴퓨터 프로젝트의 일환으로 PSI에서 실행이 어려운 대용량의 프로그램을 고속으로 처리하기 위해 개발되고 있는 기계이다. CHI의 기본 목표는 다음과 같다.<sup>19)</sup>

- 실행 성능 200K LIPS 이상
- 실기억 공간 256M.byte 이상
- 부착형(back-end) 고속 processor 구성

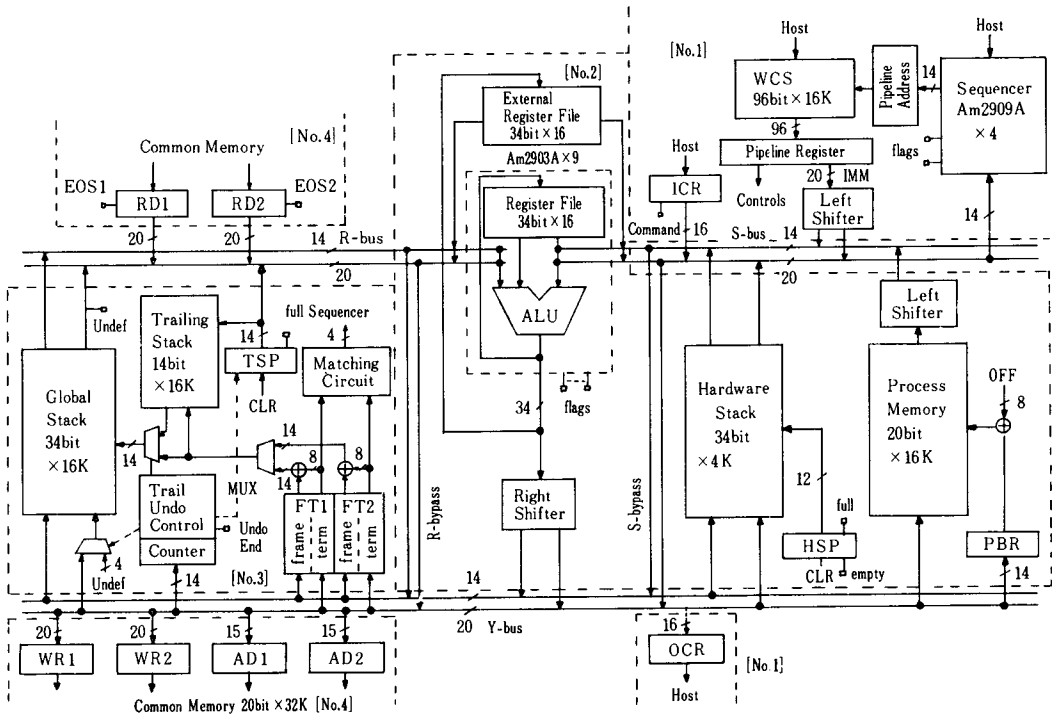


그림 3. PEK의 하드웨어 구성

200K LIPS 이상의 고속처리를 위해 특수한 하드웨어 모듈과 고급의 stack 제어 명령을 처리할 수 있는 기계구조로 설계되었으며 컴파일 코드의 최적화 기법을 도입하였다. 기계어 코드는 Warren의 코드를 기초로 하였으며 structure copying 방식을 사용하였다. CHI 시스템은 단일 사용자를 위한 기계로 CHI 프로세서, HOST 프로세서, 주기억장치로 구성된다. Host 프로세서로는 PSI를 사용하고, 대용량의 프로그램은 PSI로부터 적재되어 실행된다. (그림4)

그림 3 에서 보듯이 CHI 프로세서는 prolog processing unit, cache 기억장치, 그리고 host 프로세서 interface unit로 구성된다. CHI 프로세서는 명령어 사이클이 100ns인 마이크로 프로그램과 H/W로 제어된다.

5. Berkeley PLM(programmed logic machine)

PLM은 Berkeley 대학의 aquarius heterogeneous MIMD 머신의 co-processor로서 prolog 프로그램 수행을 위한 제어기능만 가지고 있으며, NCR/32 시스템에 부착되어 NCR/32로부터 메모리나 I/O subsystem 그리고 floating point 연산 등을 제공받는다. 그림5는 PLM과 NCR/32 시스템과의 인터페이스를 보여 주고 있다. PLM은 PLM 메모리 인터페이스인 PMI를 통해 NCR/32의 PM bus에 연결된다. PLM은 WAM에 기반

을 두고 설계되었으며, 크게 수행장치와 제어장치로 이루어져 있는데 제어장치는 마이크로 코드를 저장하는 제어기억장치와 sequencer로 구성되어 PMI 제어기와 서로 통신하면서 수행장치를 제어하고 있다. PLM의 특징은 여러 형의 자료처리를 위한 tagged 아키텍처와 하드웨어 해싱장치를 사용하는 것이다. 그리고 모든 프로세서 레지스트들을 specialized 레지스터 파일로 이루어져 여러 스택들(trail stack, global stack, local stack)을 지원하고 있다. 또 다른 특징으로, PLM은 다른 특수용도 프로세서와 연결되어 그들의 기능을 지원받을 수 있기 때문에 논리프로그램 외에도 수치계산용 프로그램도 효율적으로 수행할 수 있다. PLM의 성능은 최적화된 기계어 명령으로 컴파일하여 수행하므로 firmware 해석기에 의해 수행하는 것보다 대략 20배 정도 빠르다. 그리고 일반적인 벤치마크 프로그램으로 측정된 성능은 200~400KLIPS이다.<sup>(10)</sup>

III. 병렬 수행 추론컴퓨터(Parallel Inference Machine)

앞에서 설명했던 바와 같이 순차 수행 추론 컴퓨터의 기본적인 특징은 주어진 goal을 depthfirst 탐색과 backtracking 방식에 의해 풀어나가는 것이었으므로, 순차수행 컴퓨터는 문제를 풀어나가는 방법상의 문제점

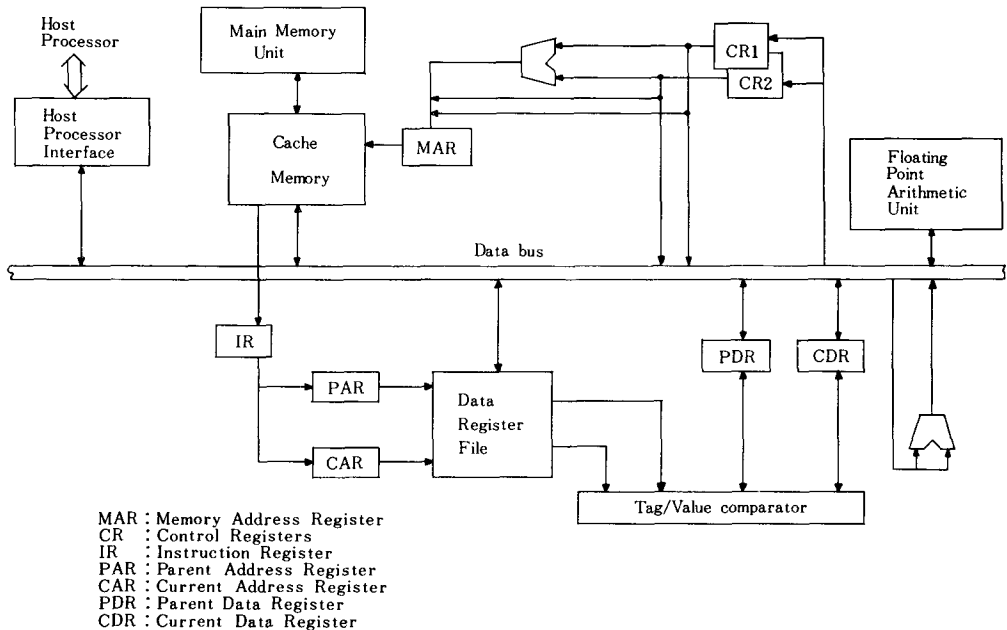


그림 4. CHI 프로세서의 하드웨어 구성

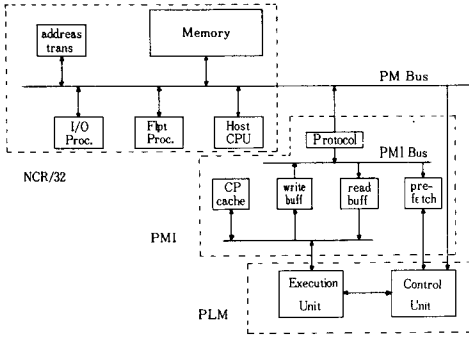


그림 5. PLM을 포함한 시스템 아키텍처

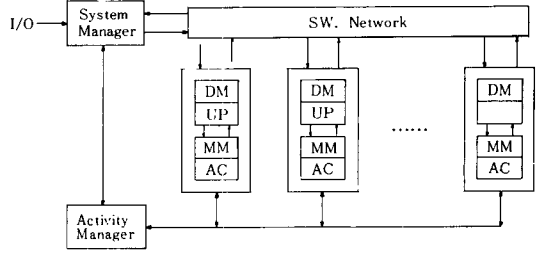


그림 6. PIE-I의 시스템 구성

으로 인해 성능 향상의 한계가 있음은 명백한 사실이다. 따라서 병렬수행 추론컴퓨터의 설계는 대용량의 논리프로그램을 고속으로 처리하기 위해 자연스럽게 추구하고 있는 실정이다. 그러나 아직은 몇가지 문제점으로 인해 실험적 단계에 머물고 있다.

논리언어의 가장 큰 특징은 비결정성(nondeterminism)이라 할 수 있으며, 이 비결정성으로 인해 언어 자체에 다양한 병렬성들이 내재하고 있고, 이들을 AND, OR, stream, backtrack, 그리고 argument 병렬성으로 나눌 수 있다. 병렬 수행 추론컴퓨터는 이들 병렬성을 활용하기 위한 수행 모델에 의해 분류하는 것이 보통이므로 이 글에서는 dataflow 모델, control flow 모델, process 모델, goal-rewriting 모델로 나누어 분류 설명한 후 각 모델을 기반으로 제작되고 있는 컴퓨터들을 소개하기로 한다.

1. Goal Rewriting 모델

Goal rewriting 모델은 논리 프로그램의 각 절(clause)이 하나의 단위가 되어 수행되는 모델로 resolution을 통해 한 개의 목적절(goal clause)이 계속해서 새로운 절로 재구성되는 방식을 의미한다. 따라서 논리언어의 여러가지 병렬수행 가능성에서 주로 OR 병렬성을 추구하고 있으며 세부적으로 unification할 때 생기는 argument 병렬성을 추구하는 경우도 있다. Goal rewriting 형식을 취급하는 것으로는 PIE(parallel inference engine)와 Haridi의 token machine, PIM-R 등이 있다.

(1) PIE

PIE는 Tokyo 대학에서 개발하고 있는 시스템으로 PIE-I 과 PIE-II가 있다. PIE-I은 그림 6와 같이 구성되는데, 한 IU(inference unit)의 구성은 프로그램을 저장하는 DM (definition memory), 끝을 저장하

는 MM (memory module), MM의 한 끝과 DM의 끝들을 단일화하는 UP(unify processor) 및 IU간의 통신을 제어하는 AC(activity controller)로 이루어진다.

AM(activity manager)은 모든 AC의 중앙 제어기이며, 시스템 매니저는 전체 PIE-I 시스템의 중앙 제어기이다. 현재 실험적으로 한 UP를 TTL과 68000 CPU를 이용하여 구현하였으며, 시뮬레이션 결과 256개의 IP를 사용했을 때 단일 프로세서보다 170배 정도의 성능이 향상되었다. PIE-II는 PIE-I을 확장한 것으로 PIE-I 과의 차이점은, 첫째로 이중 레벨의 network 구조를 가지며, 둘째로 SM(structure memory)으로 논리 언어의 structure term을 각 IM에서 공유하도록 하고, 세째로 입/출력 경로를 고려했다는 것이다. PIE-II의 전체적인 구조와 한 IU의 내부구조가 각각 그림 7과 그림 8에 나타나 있다.<sup>11)</sup>

(2) Token Machine

Haridi의 token machine 그대로 이것은 공유 메모리를 가진 다중 처리기이다. 그림 9은 token machine의 시스템 모델로 한 병렬 프로세스를 나타내는 토큰을 저장하는 token pool, 토큰이 나타내는 프로세스를

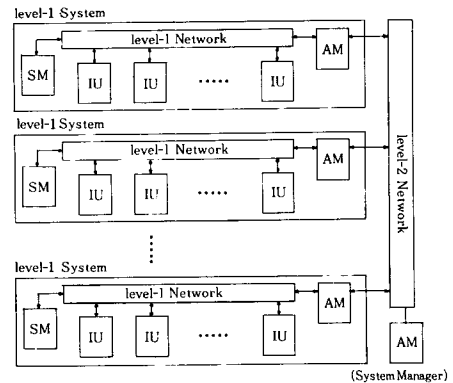


그림 7. PIE-II의 전체 구조

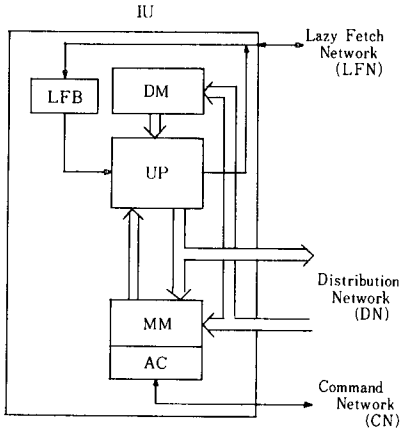


그림 8. IU의 내부 구조

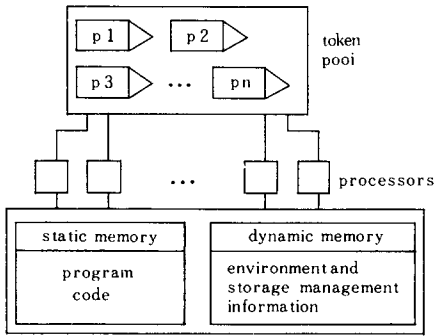


그림 9. Token Machine의 시스템 모델

풀과 단일화 장치로 구성되며, structure cope 방식을 도입하여 각 프로세스의 독립성을 유지하도록 하였다. SMM은 스트럭처 데이터를 저장하는 모듈로 모든 IM에서 공유하도록 망으로 연결되어 있다.<sup>[13]</sup> PIM-R은 reduction 모델로 분류할 수 있으나, goal rewriting 모델과 거의 차이점이 없으므로 따로 분류하지 않았다.

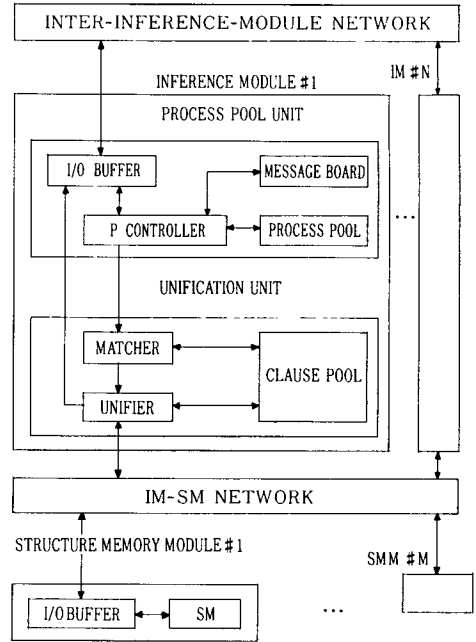


그림 10. PIM-R의 개념적인 구조

수행하는 프로세서들, 그리고 프로그램 코드와 environment 및 그의 제어정보를 저장하는 공유 메모리를 보여 준다. 현재 프로세서의 machine cycle 및 인스트럭션들이 정의되어 있는 상태이다.<sup>[12]</sup>

(3) PIM-R

PIM-R은 ICOT에서 개발하고 있는 시스템으로 두 가지 언어(prolog와 concurrent prolog)로 된 프로그램을 수행하도록 되어 있다. 이는 수행시 AND 병렬성을 수행하기 위한 부담을 줄이기 위한 것으로, AND 병렬성을 추구하기 어려운 일반 prolog 프로그램은 OR 병렬성만 추구하며, concurrent prolog 프로그램에 대해서는 AND 병렬성만을 수행한다. PIM-R의 개념적인 구조는 그림 10에 보이는 바와 같이 두가지 타입의 모듈, 즉 IM (inference machine)과 SMM (structure memory module), 그리고 그들을 연결하는 망으로 이루어진다. IM은 프로세스를 수행하는 모듈로 프로세스

2. AND/OR 프로세스 모델

AND/OR 프로세스 모델 (이하 프로세스 모델)은 하나의 목적절(goal clause) 전체를 수행 단위로 사용하는 goal-rewriting 모델과 달리 목적절을 부 목적절(sub-goal)로 수행 단위를 나누어서 각각을 개별적인 프로세스에 할당하여 풀어나가는 방식이다.<sup>[14]</sup> 프로세스 모델은 논리 프로그램의 병렬 수행 가능성 중에서 AND/OR 병렬성을 동시에 추구할 수 있는 모델로서 다른 모델보다 이론적으로 완벽하며 한 프로세스가 지역적인 정보만을 가지고 수행하게 되므로 분산처리가 용이하다. 그러나 공유변수로 인해 프로세스 사이의 동기 맞추기가 어렵고 프로세스의 숫자가 기하 급수적으로 불어나게 되며, AND 병렬성을 위해서 수행시에 과도한 부담이 생기게 된다. 따라서 프로세스 모델을 실제 수행모델로 사용하기 위해서는 여러가지 기술적인 문제점을 안고 있다.<sup>[15, 16, 17]</sup>

아직은 프로세스 모델을 이용한 본격적인 병렬 추론 컴퓨터에 대한 발표는 되고 있지 않으나 가장 활발한 연구활동이 전개되는 부분이다. 프로세스 모델을 이용한 병렬 추론 컴퓨터로는 유일하게 Kobe 대학의 PARK 가 있는데 PARK는 공유기억장치를 가지는 버스 시스템에 다수의 68000 프로세서로서 구성된다. 각 프로세서는 LM (local memory), ADT (address translation unit) 그리고 CM (common memory)로 구성되며 각 프로세서 사이의 메시지 교환은 CM을 통하여 이루어진다. PARK는 AND/OR 병렬성을 동시에 추구할 수 있으며 논리언어에 프로그래머가 제어정보를 줄 수 있는 PARK-prolog를 사용자 인터페이스로 이용한다(그림11). 현재는 프로세스 사이의 동기를 맞추는 방법과 프로세스의 숫자를 제한하는 방법, 그리고 수행시 프로세스의 부담을 줄이기 위한 방법들이 계속해서 발표되고 있으며 앞으로 이를 기초로한 병렬 수행 추론 컴퓨터들이 계속 발표될 것으로 생각된다.

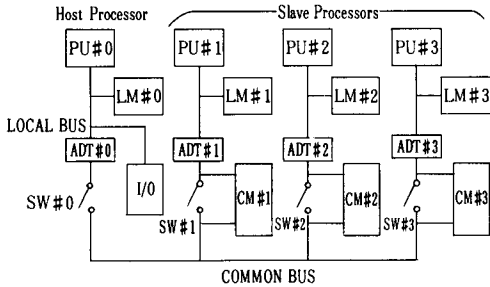


그림11. PARK의 하드웨어 구성

3. Dataflow 모델

논리 프로그램을 위한 dataflow 모델은 기존의 dataflow 개념을 논리 프로그램의 병렬수행 모델과 접목시킨 것으로, 논리 프로그램을 dataflow 그래프로 변환시킨 후 기존의 dataflow의 제어방식을 적용하는 모델이다(그림12). Dataflow 모델은 논리언어의 병렬성을 추구하기 위한 새로운 방법으로 제시되고 있지만 논리언어가 가지는 비결정성(nondeterminism)을 얼마나 효과적으로 수행할 수 있는가는 아직 의문점으로 남아 있다. Ito의 PIM-D는 이러한 dataflow 모델을 기반으로 하는 병렬수행 추론 컴퓨터로 그림13와 같이 여러 개의 structure memory들과 프로세서들로 연결되어 있다. PIM-D에서는 AND/OR 병렬성과 단일화(unification) 작업에서 이루어지는 인수 병렬성(argument

parallelism)을 수행할 수 있도록 설계되어 있으며, 여러가지 데이터 형(type)을 구별하기 위해 tag를 사용하는 방식을 취하고 있다. Structure memory에는 리스트나 벡터와 같은 structre형의 데이터가 분산저장되어 프로세서들 사이에 공유된다. PIM-D는 실험적인 형태로 구현될 예정으로 아직 구체적인 성능은 알려지지 않고 있다.<sup>12,13)</sup>

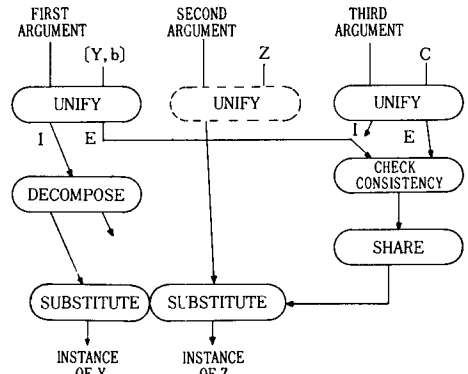


그림12. 절 p((Y, b), X, c) : -의 Dataflow 그래프

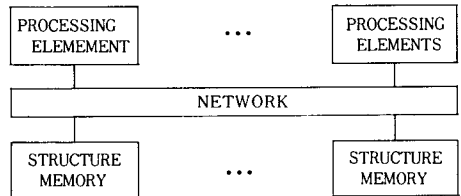


그림13. PIM-D의 개념적 구조

4. Control flow 모델

Control flow 모델은 다른 병렬 수행 모델들과 달리 순차 수행 컴퓨터를 다중 프로세서로 확장시키는 방식을 취하고 있다. 대표적인 control flow 모델은 텍사스 대학의 RAP-WAM (restricted and parallelism-Warren abstract machine)과 일본 Oki 회사의 ORBIT 가 있다. RAP(restricted and parallelism) 개념을 도입하고 있다.<sup>14)</sup>

RAP-WAM에서는 한개의 프로세서는 WAM 형식으로 수행되며 프로세서 사이에는 메시지 교환을 통해 병렬 수행을 할 수 있도록 프로세스 모델을 채용하고 있다. 즉, 실제 수행모델은 WAM이며 전체 다중 프로세서의 제어 모델은 프로세스 모델을 사용한다. RAP

표 3. 순차 수행 추론컴퓨터의 비교표

특 성 순차 수행 추론컴퓨터	기 계 어	수 행 속 도	현 재 상 태	Structure Sharing /copying	특수 H/W	Cache
CHI	Warren의 기계어 변형	약 200KLIPS	완 성 (?)	Copying	H/W Stack	yes
PSI-I	KL $\phi$	30KLIPS	완 성	Sharing	no	yes
PSI-II	KL $\phi$	약 100KLIPS	제 작 중 (87년 보완성)	Copying	no	yes
PEK	Prolog	약 40KLIPS	제 작 중	Sharing	H/W Stack	no
Pipelined-prolog machine	Warren의 기계어	200~450 KLIPS (모의실험)	제 작 중 (?)	Copying	Pipelined E-unit, I-unit	no
Berkeley PLM	Warren의 기계어 변형	200~400 KLIPS (모의실험)	제 작 중	Copying	H/W Hashing	no

cf : \*DEC 2060 compiler 43KLIPS  
 \*\*C-prolog Interpreter 1.5KLIPS

표 4. 병렬 수행 추론컴퓨터의 비교표

특 성 병렬수행 추론컴퓨터	시스템 구조	수 행 모 델	병 렬 성	상 태	논리언어 인터페이스
PIM-R by ICOT, Japan	Loosely coupled multiprocessor	Reduction model	AND/OR 병렬성, 인수병렬성	제 작 중	Prolog Concurrent Prolog, Guarded Horn Clause
PIM-D by ICOT, Japan	"	Dataflow model	"	"	Parallel Prolog, Concurrent Prolog
PIE by University of Tokyo	Loosely coupled multiprocessor/multicomputer	Goal-rewriting model	OR 병렬성, 인수병렬성	"	Pure Horn Clause
PARK by Kobe University, Japan	Bus-based multiprocessor with master-slave control	Process model	AND/OR 병렬성	제 작 완 성	PARK-Prolog
RAP-WAM by Texas at Ostin U. S. A	not yet	Control flow model	AND 병렬성	제 작 중	Prolog
ORBIT Oki Co., Japan	Loosely coupled multiprocessor with central control processor	"	OR 병렬성	"	Prolog

이러한 프로세스 모델에서 생기는 과중한 수행 부담을 줄이기 위해 AND 병렬성을 희생하면서 수행 부담을 줄이는 컴파일 방식을 의미한다. ORBIT 역시 WAM

을 다중 프로세서로 확장하기 위한 모델이다. RAP-WAM과 차이점은 ORBIT는 RAP을 사용하지 않으며 수행시에 병렬로 수행 가능한 부분을 프로세스 bundle



이라 불리우는 단위로 나누어 다른 프로세서에 이전하는 방식을 사용한다. Bundle의 이전은 전체 프로세서들을 관리하는 한개의 제어 프로세서에 의해 이루어진다. RAP-WAM과 ORBIT는 아직 실험단계에 머물고 있으며 실험적인 제작이 진행되고 있다.<sup>[19]</sup>

#### IV. 결 론

앞에서 요약한 바와 같이 순차 수행 추론 컴퓨터는 논리 프로그래밍 환경조성을 위해 빠른 속도로 개발이 추진되고 있다. 표 3에서 보듯이, 순차 수행 추론컴퓨터의 성능은 기계어로 prolog를 직접 수행하는 경우보다 특정한 기계어 코드로 사용할 때가 훨씬 좋은 것을 알 수 있으며 structure를 sharing하지 않고 copying 하여 사용하는 추세이다. 현재 순차 수행 추론컴퓨터의 성능은 최고 200~400KLIPS로 되어 있으며, 앞으로 H/W unification unit나 H/W stack 등의 특수한 하드웨어 설계기술과 소자기술 개발에 따라 더 향상될 전망이나 MLIPS의 성능을 위해서는 아직도 상당기간이 필요한 것으로 예상된다. 병렬 수행 추론컴퓨터를 특성에 따라 분류하면 표 4과 같다. 표에서 보듯이 제작이 완성되어 있는 컴퓨터는 PARK에 불과하며 거의 실험적인 상태에 머물고 있으므로 구체적인 성능 평가는 물론 상대적인 성능 평가조차도 할 수 없는 상태이다. 단지 수행 모델과 추구하는 병렬성들로만 분류 가능할 뿐이다. 이러한 병렬 수행 추론 컴퓨터의 실용화를 위해서는 수행 모델을 좀 더 구체화 시키는 문제이외에도 자원의 공유 문제, 연결망의 성능 문제, 그리고 job 분산처리 문제 등 기존의 다중처리기에서 생기는 여러 문제들에 대한 고려가 병행되어야 할 것이다.

#### 參 考 文 獻

[1] K. Hwang, J. Ghosh and R. Chowkwanyun, "Computer Architectures for Artificial Intelligence Processing," *IEEE Computer*, vol. 20, no. 1, pp. 19-28, 1987.

[2] K. Murakami et al., "Research on Parallel Machine Architecture for Fifth-Generation Computer System," *Computer*, vol. 18, no. 6, pp. 76-92, June 1985.

[3] E. Tick and D.H.D Warren, "Towards a Pipelined Prolog Processor," *New Generation Computing*, vol. 2, pp. 323-345, 1984.

[4] M. Yokota et al., "The design and implementation of a Personal Sequential In-

ference Machine: PSI," *New Generation Computing*, vol. 1, no. 2, 1983.

- [5] S. Uchida, et al., "Outline of the Personal Sequential Inference Machine," *New Generation Computing*, vol. 1, 1983.
- [6] K. Fuchi et al., "The Role of Logic Programming in the Fifth Generation Computer Project," Key note of 3rd Int. Conf. on Logic Programming, pp. 1-24. 1986.
- [7] N. Tamura et al., "Sequential Inference Machine PEK," Proc. of FGCS84, pp. 542-550. 1984.
- [8] D.H.D. Warren, "An Abstract Prolog Instruction set," Tech. Report 309, Artificial Intelligence Center, SRI International 1983.
- [9] R. Nakazaki, "Design of a co-operative high performance sequential inference machine (CHI)," NEC R
- [10] T.P. Dobry et al., "Performance study of a prolog machine architecture," 12th annual Symposium on Computer Architecture, pp. 180-190.
- [11] A. Goto, H. Tanaka and T. Moto-Oka, "Highly parallel inference engine PIE-goal rewriting model and machine architecture," *New Generation Computing*, vol. 1, pp. 37-58. 1984.
- [12] A. Ciepielewski and S. Haridi, "Execution of bagof on the or-parallel token machine," Proc. of FGCS'84, pp. 551-562, Nov. 1984.
- [13] N. Ito, H. Snimizu, M. Kishi, and K. Rokusawo, "Data-flow based execution mechanisms of parallel and concurrent prolog," *New Generation Computing*, vol. 2, pp. 37-58, 1984.
- [14] J. Concery and D.F. Kilber, "AND parallelism and Nondeterminism in Logic Programs," *New Generation Computing*, vol. 3, pp. 43-70, 1985.
- [15] D. Degroot, "Restricted and-parallelism," Proc. of FGCS'84, pp. 471-478, Nov. 1984.
- [16] H. R. Lee, S.B. Kim, S.R. Maeng and J.W. Cho, "A parallel execution model of logic programs on tree structured multiprocessor," Proc. of SCS'85, pp. 65-72, Dec. 1985,

[17] S.B. Kim, S.R. Maeng and J.W. Cho, "A parallel execution model of logic programs based on dependency relationship graph," Int. Conference on Parallel Processing, Aug., pp. 976-983, 1986.

[18] M.V. Hermenegildo et al., "An abstract machine for restricted AND-parallel execution of logic programs," 3rd Int. Conference on Logic Programming, pp. 25-39, 1986.

[19] H. Yasuhara and K. Nitadori, "ORBIT: A parallel computing model of prolog," New Generation Computing, vol. pp. 277-288, 1984. \*

♣ 用語解説 ♣

**Asynchronous (비동기)**

컴퓨터가 동작되는 한 형태로 이전 명령의 동작이 끝났다는 신호가 와야 다음 명령의 수행이 시작되는 것

**Baud (보드)**

원래 전보 전송에서 사용되던 기술 용어로서 단위 시간당 변조(modulation)율을 나타낸다. 현재 컴퓨터 분야에서는 데이터의 흐름을 나타내는 단위로 쓰이며, 초당 전송되는 비트의 수를 의미한다. 보드(J. M. E. Baudot)라는 사람의 이름에서 유래되어 명명되었다.

**Bifurcation (이진 논리조건)**

단지 두 개의 상태만을 가질 수 있는 논리조건. 이것은 바로 이진법 디지털 컴퓨터의 기본 논리이다.

**Bubble Memory (버블 메모리)**

자기 물질막(a film of magnetic material)에 산재되어 있는 조그만 원통 모양의 자성버블(magnetic bubble)을 기초로 한 기억장치. 이 버블은 주변의 자성 물질과는 반대의 극성을 갖는다. 버블을 만들고 없애며 또 이리저리 움직이도록 하는 방법이 개발되었다. 버블이 있고 없으므로 해서 정보를 나타내며 버블을 이동시켜 디지털 정보를 조작할 수 있음. 자기 버블 기억장치(magnetic bubble memory)는 신뢰도가 높고 속도도 비교적 빠르며, 부피, 무게, 전력소모등이 작아 종래의 기억 장치에 비해 많은 이점을 가지고 있다.

**Cache (캐쉬)**

주 기억 장치에 읽어 들인 명령이나 프로그램들로 채워지는 버퍼 형태의 고속 기억장치. 수행시 필요로 하는 명령어나 자료가 캐쉬에 있으면 고속으로 수행이 가능하고 캐쉬에 없으면 주 기억 장치에서 새로 읽어 들인다.

**Compile (컴파일)**

원시 프로그램을 기계어 프로그램으로 번역하는 작업. 변환된 목적 프로그램에서 부프렸그램들은 각각 연결되어 있으며 기억 장소에 적재(load)하여 바로 수행할 수 있다.

**Computerized Numerical Control (CNC) 컴퓨터 수치제어**

기본적인 수치제어 기능을 수행하기 위해 프로그램 내장 방식의 전용 컴퓨터를 사용하는 수치제어 시스템

**Firmware (펌웨어)**

기계의 일부분을 구성하는 제어장치에 쓰여진 하급의 컴퓨터 프로그램으로 소프트웨어와는 달리 펌웨어는 컴퓨터의 전체 조직과 관련이 있는 프로그램이다.

**Form Stop (움지 정지)**

인쇄 장치에서 종이가 없으면 자동적으로 동작을 멈추는 장치