

C-element를 이용한 Node Label Data Flow Machine 설계

(Design of a Node Label Data Flow Machine Using C-elements)

김희숙* · 조현섭**

(Hee-sook Kim · Hyeon-seob Cho)

1. 서론

Data Flow Machine들은 다른 병렬 컴퓨터구조와 다르다. 프로그램 내장방식의 기존 sequential 및 synchronous 방식 컴퓨터와는 달리 operand data의 유무가 실행 여부를 결정하는 Data Driven, Asynchronous 방식의 병렬처리 장치이다[4][5].

Data Flow Machine의 개념에 상응하여 여러 가지 모델들이 발표되었으나 본 논문에서는 Davis의 Data Flow Machine 요구사항에 만족하고, Dennis와 Manchester 모델에 근거한다. machine code를 local memory에 내장시켜 main memory의 access 부담을 줄이고, 단순 구조, 단순 동작 논리를 갖는 processing site를 구성한다. 병렬 연결에 의해 시스템 확장이 가능하도록 하였고, 전체적으로 Recursively Hierarchical structure를 이루는 시스템을 구성하였다.

Data Flow Machine에서 임의의 node에 데이터 도착 여부는 firing의 순간을 결정하고, 동작속도와 동기화를 위해서 중요한 의미를 갖는다. 따라서 본 논문에서는 기 설계된 Node Label Data Flow Machine에 기초하여 Muller의 C-element를 적용함으로써 Self-timed 방식에 의한 동기화 문제를 고찰한다.

2. 본론

2.1 Node Label Data Flow Machine 구조

2.1.1 전체적인 시스템 구조

Data Flow Machine은 out-of-order 명령을 실행하기 위하여 현재 RISC machine과 같은 복잡한 H/W나 S/W 기술없이 concurrent하게 데이터를 inherit하여 연산을 수행한다. 따라서 노이만식에 비해 병렬처리성이 높고, 구조 및 동작의 단순화, pipelining의 적용성, 확장성등의 장점을 갖는다.

Node Label Data Flow Machine은 Davis의 Data Flow Machine 요구사항에 만족하고, Dennis와 Manchester 모델에 근거하여 설계하였다. 시스템은 Data Flow Graph와 Node Label을 이용하여 Node token이 실행되도록 구성하였다. 전체적인 시스템의 구성으로 processing site는 체계적 또는 단계적 집합으로 구성하였고, 각 processing module은 local memory를 가지며, 각 PM내의 명령 실행은 단순하다.

그림 1은 3부분으로 구성된 전체적인 시스템이다. processing site인 PM들은 병렬로 연결되어 있고, 각 PM에서 생성된 result token을 해당 목적지로 분배하는 Distributer가 있으며, 입출력 제어부분으로 구성된다.

2.1.2 명령어 형식

Node Label은 임의 Node의 순차순서(i)와 병렬순

* 정희원:원광대학교 공과대학 컴퓨터공학과 박사과정

**정희원:청운대학교 공과대학 전자공학과 전임강사

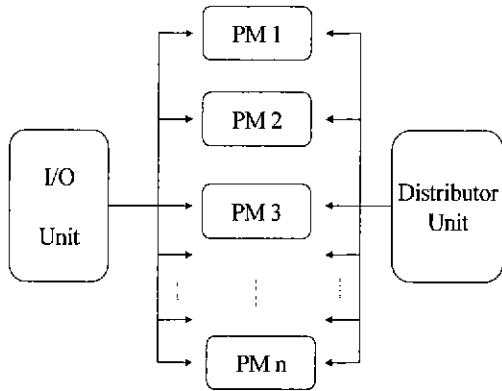


그림 2. 전체적인 시스템 구성도

서(j)을 나타내는 2차원 첨자로 구성된다.

$$\begin{aligned}
 NL &:= i, j \\
 i &= 1, 2, 3, \dots, S_{max} \\
 j &= 1, 2, 3, \dots, P_{max}
 \end{aligned}$$

i는 데이터의 의존도를 나타내고, j는 병렬처리의 동시성을 나타낸다. S_{max}는 순차처리시 최하위 순서가 되고, P_{max}는 시스템의 총 프로세서수가 된다. 모든 명령 packet은 해당 노드의 결과에 대한 목적지 주소(arc)를 자체에 포함하고 있다. 따라서 source arc의 코딩을 할 필요가 없다.

모든 node들의 실행을 위해 필요한 명령어 형식은 그림2와 같다.

Node Token	Operand Data	Control Data	Flag
------------	--------------	--------------	------

그림 2. 명령어 형식

operand data와 control data는 경우에 따라 1개 또는 2개가 된다. Flag의 값에 따라 데이터의 유무를 판단한다.

모든 명령은 Node Token Memory에 보관되고, 명령어 형식에 따라 PM의 내부를 다음과 같이 구성한다.

2.1.3 PM의 구조

각 PM의 구조는 그림 3과 같다. Node token, operand, ETF값을 보관하는 local memory와 processing unit 및 distributor를 위한 레지스터들로 구성되어 있고, ring type의 실행 사이클을 갖는다.

각 장치들의 기능으로 NTM은 op code와 1개 또는 2개의 목적지 NL값을 보관하고, ETF로부터 enabled token의 주소를 공급 받는다. DTML/R은 NTM과 동일 주소에 위치한 Node가 사용할 operand 데이터를 보관하는 메모리로서 ETF로부터 addressing된다. CF는 control Node가 필요로 하는 제어용 데이터를 보관하는 1비트 메모리이다. ETF는 Node의 실행 가능 여부를 표시하는 flag로서 L/R/C 값은 각각 DTML/R 및 CF내의 데이터가 도착하였는지의 여부를 나타낸다. 결과의 read/write 과정마다 ETF=111인지를 체크한다. PU는 Node token의 op code를 decode하고, operand 데이터에 대하여 산술, 논리, 제어용 operator를 내장한다. 실행결과는 Distributor로 보내져서 목적지 주소로 보내진다. 시스템의 제어는 전체적인 system clock에 의존하지 않고, 분산된 clock을 사용하며 각 장치는 local clock을 사용한다.

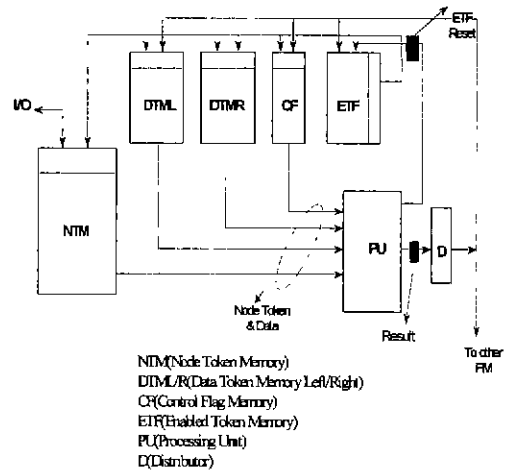


그림 3. PM의 기본 구조

2.2 C-element를 이용한 Node Label Data Flow Machine 설계

2.2.1 Self-timed Circuit

Self-timed circuit은 전역 클럭을 사용하지 않고 시스템 구성요소들 사이의 특정 프로토콜에 의하여 동작이 제어되므로 VLSI소자를 더 효율적으로 만든다. 또한 초기의 비동기 회로와 달리 speed-independent한 설계를 할 수 있다.

Self-timed system은 Data 채널, request 채널, Acknowledge 채널로 구성되고, 모듈 사이의 비동기 통신 프로토콜을 사용한다.

모듈사이의 통신 기법은 2-cycle과 4-cycle이 있다. 2-cycle은 유효한 데이터만 전달하므로 4-cycle보다 속도는 빠르고 전력도 덜 소모하지만 회로가 커질수 있는 단점이 있다. 본 논문에서는 각 arc에 데이터가 도착한 순간 firing을 해야하므로 2-cycle 기법을 사용했다.

그림 4는 하나의 Self-time system 모듈 구성도이다.

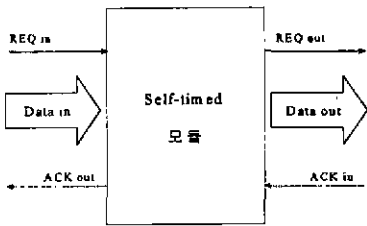


그림 4. Self-timed 모듈

2.2.2 구현

그림 5는 Self-timed 모듈로 구현된 PM 구조의 블럭도이다.

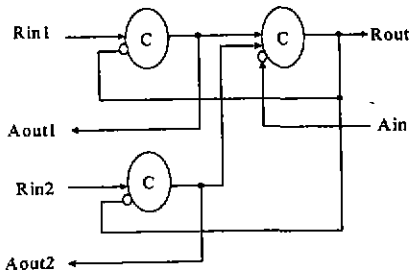


그림 6. PU에 적용된 C-element

Local memory와 DTML/R, CF을 MU블럭으로 만들고, PU와 D로 구성되며, 파이프라인 구조를 이룬다.

그림 6은 Muller의 C-element를 적용한 PU의 한 부분이다. RR1과 RR2의 결과 토큰이 Distributer의

입력 토큰을 위해 사용된 bubbled Muller C-element이다. Muller의 C-element를 이용하여 Self-timed circuit을 구현한 이유는 Speed-independent 즉, hazard-free 때문이다. PM은 5단계로 모듈화시킬수 있고, 2-cycle에 의해 동기화를 이룬다.

실험을 통하여 각 arc에 데이터가 도착하여 firing할 수 있는 순간을 ETF가 111일때의 조건 보다 정확함을 확인할 수 있었다.

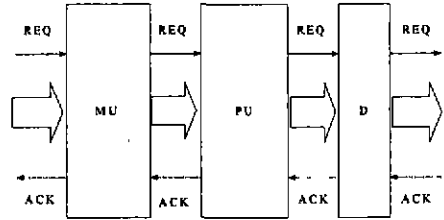


그림 5. Self timed 모듈로 구현된 PM 구조

3. 결론

본 논문에서는 Muller의 C-element를 이용한 Node Label Data Flow Machine을 설계했다.

Data Flow Machine은 임의노드에 데이터 도착여부의 판단을 어떠한 방법으로 활용할 것인가가 중요한 의미를 지닌다. 노드들은 항상 자신의 모든 입력 단자에 필요한 모든 데이터 비트들이 도착할때만 firing을 하는데, 어느 시점에서 이 모든 데이터들이 도착되었음을 노드에게 신호해줄것인가의 문제는 동작 속도와 동기화 문제에 있어서 중요하다.

본 논문에서는 Self-timed 회로로 설명되는 Muller의 C-element를 이용하여 기존의 Node Label Data Flow Machine에 적용시킴으로써 ETF를 이용할때보다 정확한 firing순간을 확인할 수 있었다.

참고문헌

- (1) Shih-Len Lu, Chih-Ming Cahng, "Modeling of a Selftimed Dataflow Processor in VHDL", Proc 6th IEEE ASIC Conference, 1993
- (2) Ilana David, Ran Ginosar, Michael Yoeli, "Implementing sequential machines as self-timed circuits", IEEE transactions on computers, VOL 41,NO.1, 1992,
- (3) G.Theodoropoulos, J.V.Woods, "Distributed simulation of asynchronous computer architectures:The program driven approach"
- (4) Arvind, Agerwala, "Data Flow system", 1982, computer
- (5) Guang R.Gao, Gabriel Musilberman, "Parallel Architectures and Compilation Techniques", MICEL, 1994