

# 世界 最初の 非 Neumann型 Computer

## — 第5世代 Computer 開發의 실마리 —

世界 최초의 非 Neumann型 超高速 Computer의 實用化가 話題를 불러 일으키고 있다. 83年 11월에 東京에서 개최되었던 國際寫眞測量學會에서 日本電氣가 발표한 非Neumann型 Computer는, 현재의 大型 Computer와 비교할 경우 50~100배나 빠른 53 MFLOPS (1秒에 5,300만회)의 超高速 計算을 할 수 있는 것이다. 아직까지는 資源 探查衛星으로부터 送出되는 畫像處理 專用に 한정되어 있으나, 先進 諸國이 연구 개발에 총력을 기울이고 있는, 人工知能을 지닌 第5世代 Computer 開發에의 突破口 역할도 하는 것으로 기대되고 있다.

이하, 이 革新的인 Computer에 대해서 살펴 본다.

### 1. Software 危機의 打開

#### 가. Neumann型 Computer

1945年, 美 펜실바니아大學의 Von Neumann이, Memory Cell 과 Program Counter의 存在를 특징으로 하는 Computer, Architecture (非 Neumann型 Computer, Stored Program 方式)을 提案한 이래 현재까지 세계에서 稼動되고 있는 대부분의 Computer는, 그 Architecture에 기반을 두고 있다. 이러한 Neumann型 컴퓨터는, 圖1에서와 같이, 制御裝置(CU)·演算裝置(ALU)·記憶裝置(M) 및 入出力裝置(I/O)를 그 기본 構成 요소로 하고 있다.

記憶裝置에는, 처리해야 할 Data와 처리 내용을 記述한 Program이 축적되어 있다. 制御裝置는, 同裝置 内に 있는 Program Counter (다음에 實行해야 하는 命令이 저장되어 있는 番地를 나타내고 있다)에 의해서 하나씩 순번대로 記憶裝置에서 명령을 취하여 그 명령대로 記憶裝置 内の Data에 대해서 演算裝置로 연산하거나, 연산 결과를 기억하거나 出力하는 등 Control을 행한다.

이와 같이 Neumann型 컴퓨터에서는, Program Counter에 의해 處理 Sequence가 결정되는 것으로, Program은 Single Control Flow Stream에 따라 만들어지지 않으면 안 되며 기본적으로는 命令 레벨의 Parallel 처리는 할 수 없다.

또, 처리 Data, 制御 Data, Address Data 등 많은 Data가 制御裝置와 記憶裝置를 결합한 하나의 길(Tube)을 통하기 때문에, Data 轉送에서 Battle Neck가 생길 가능성이 있다.

그리고 Neumann型 컴퓨터用 Software (Fortran Cobol 등)에 대해서는 問題 記述의 어려움, 復雜性, 記述 능력의 미약, Program Test의 어려움, Parallel 처리의 곤란 등 때문에 「Software 危機」가 사망에서 문제시되고 있다.

이러한 危機의 원인이 현재 Neumann型 컴퓨터의 Architecture 및 그것에 기초한 Programming 言語로 있는 어떤 認識으로부터 非Neumann型 컴퓨터가 활발하게 연구되고 있다.

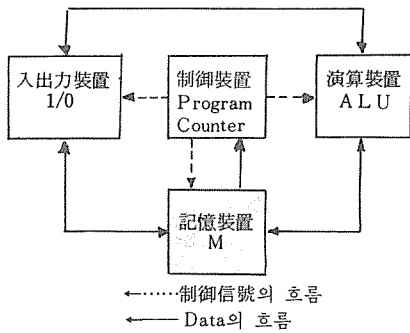


圖 1 Neumann型 Computer의 기본 構成

## 2. Ring 構成의 Data 驅動型

日本電氣의 非Neumann型 컴퓨터는 Ring 構成의 Data 驅動型 Data Flow Machine이며 그 Program 言語는 關數型 言語이다.

Data Flow Machine의 動作은 Data Flow Graph를 사용해서 할 수 있다. 圖 2는 Data Flow Graph의 例이며, Data의 흐름을 나타내는 Arc, Data의 複製를 나타내는 Link Node, 갖가지의 演算을 나타내는 Node를 이용해서 나타나 있다.

演算은, 그 연산에 필요한 Data가 갖추어지면 곧 實行된다. 圖 2에 나타나 있지만 演算 Node MUL의 실행은 프로그램의 記述 순서에 상관하지 않고 Data X1과 X2가 MUL에 도착하면 개시된다.

圖 2에 演算 내용을 關數型  $\langle y=f(X1, X2 \dots) \rangle$ 의 形 언어로 기술한 例를 표시하였다.

일반적으로 Data Flow Machine用의 프로그램 言語는 단일 代入 規則 (Single Assignment Rule)에 의하지 않으면 안 된다. 단일 代入

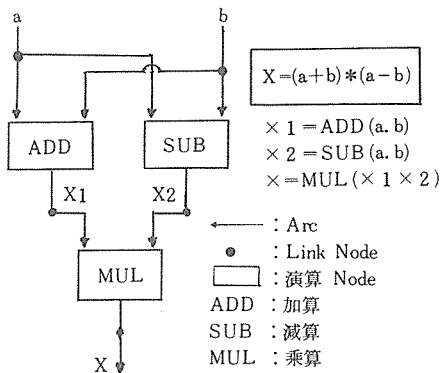


圖 2 Data Flow Graph의 例

規則은 演算式의 左邊(演算 結果)에 同一의 變數名이 2度 이상 나타나서는 안 된다고 하는 規則이며, 이 制限을 마련함에 따라, 1度值의 定해진 變數는 그 후 변하지 않았다.

환언하면, 프로그램에서 文의 순서를 변경하더라도 演算 結果에 영향을 주지 않는다. 이와 같은 關數型 言語를 실현하는 Data 驅動型의 Data Flow Machine은 演算은 그 關數(ADD, SUB, MUL 등)에 필요한 Data(入力 Token)가 정리되면 즉시 실행되기(Data 驅動의 原理) 때문에 圖 2의 例에서는 加算과 乘算이 동시에 실행된다. 이러한 Data 驅動型 Machine은 본질적으로 Parallel 처리 能力을 갖추고 있다.

이상 살펴본 바와 같이, Data Flow Machine(非 Neumann型 컴퓨터)에는 Program Counter 및 Memory Cell이라고 하는 概念은 존재하지 않는다.

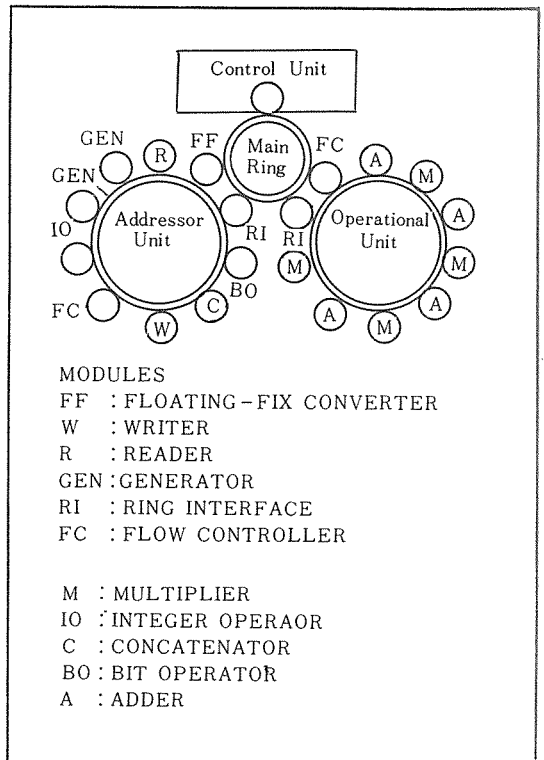


圖 3 NEC가 개발한 Data 驅動型 Data Flow Machine(DFM)의 構成(DFM(非 Neumann型 Computer)는 本質的으로 Parallel 處理能力을 갖추고 있으며, Program Counter 및 Memory Cell의 概念은 없다.)

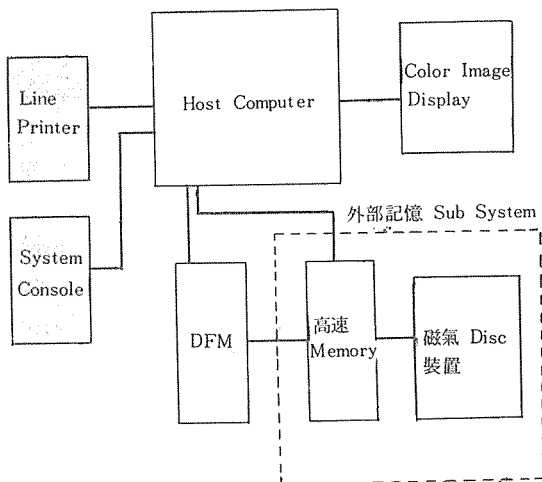
圖 3에日電이 개발한 Data 驅動型 Data Flow Machine의 구성을 나타냈다.日電의 Data Flow Machine (DFM)은, Control Unit, Main Ring, Addressor Unit, Operational Unit로부터 이루어져 있다.

Ring 狀으로 결합된 複數 개의 처리 Module 은, Data 驅動制御에 의해 동작한다. Ring 狀 Bus를 통하는 Data는, 數值만이 아닌 Tag(變數名)을 가지고 있고, Data는 Tag로 지정되는 처리 Module에 자동적으로 흘러 들어가 처리에 필요한 Data가 정리되면 즉시 실행된다.

Control Unit는, 16Bit의 Micro Computer로 각 Unit의 Initialise 등의 制御를 행한다. Addressor Unit는, Reader와 Writer에 에 접속된 Memory에 대한 Address 발생과 Reader·Writer를 매개체로 한 Memory와의 Data 授受, Main Ring 經由로 Operational Unit에 Data 供給, Operational Unit의 演算 결과의 정리 등을 행한다.

Operational Unit는 受取한 Data를 처리하여 Addressor Unit로 보낸다. Main Ring 은 이러한 3개의 Unit의 Interface 동작을 한다.

이러한 構成을 갖는日電의 DFM은, 처리 Module의 변경, 추가 및 Multi Processor 化를 비교적 용이하게 하는 일을 할 수 있으므로, 요구되는 처리에 最適한 시스템을 구축할 수 있다.



寫眞 1 NEDIPS Hardware의 構成

### 3. 1 Scene의 SAR 畫像이 3時間으로

日電은, 畫像 Data의 高速處理를 목적으로 Data Flow Machine을 核으로 하는 高速畫像處理시스템 NEDIPS (NEC Data Flow Image Processing System)을 개발하여, 畫像 처리 중에서도 아주 처리 Data 量이 많은 合成開口 Radar (SAR)의 Data 처리에 적용하였다.

SAR은, Remote Sensing用 Micro波 Sensor의 一種으로 人工衛星 등으로부터 夕陽을 가리지 않고 天候에 좌우되는 일 없이 地球 표면의 모습을 상세하게 觀測하는 일을 수행할 수 있는 우수한 Radar이지만, 再生 畫像을 얻기 까지에는 여러 가지 복잡하며 大量의 처리를 요구한다.

汎用 大型 컴퓨터를 이용한 처리에는 1 Scene의 SAR 畫像을 얻기 위해서는 수백 시간을 요하지만, NEDIPS를 이용해서 처리할 경우 3~4시간으로 얻을 수도 있는 것이다.

NEDIPS는, 圖 4에서와 같이 Host Computer (Mini Computer NEC MS 50) DFM 및 外部 記憶 Sub System으로부터 성립되었다.

Host Computer는, System 전체의 管理, 즉 DFM과 外部 記憶 Sub System의 管理 및 Color Image Display를 포함한 MS 50의 周邊機器 管理를 행한다. DFM은 Host Computer의 管理下에, Data Flow에 의한 Data 처리를 實行한다.

외부 記憶 Sub System은, 大容量 高速 IC Memory와 磁氣 Disc 裝置로 구성되고, 高速 Memory는 DFM의 Data Memory 및 Host Computer와 DFM의 Buffer Memory로 동작한다. 또한 磁氣 Disc는 처리 Data의 일시 記憶 File로써 사용한다.

高速 Memory에는 複數의 Data Bus가 있는데 DFM 및 磁氣 Disc를 독립으로 Access 할 수 있으며, DFM의 高速性を 유효하게 만드는 일도 할 수 있다.

SAR의 Data 처리는 Range 壓縮·Azimuth 壓縮이라고 불리어지는 2 종류의 相關演算과 Corner Turning이라고 불리어지는 Data의 줄 교환 處理 등으로 구성된다.

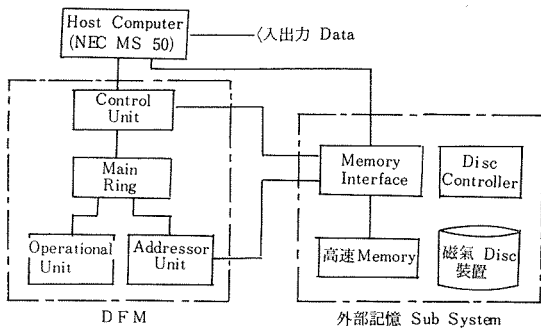


圖 4 NEDIPS Block 圖

圖 5 는 NEDIPS에서의 SAR Data 처리 흐름을 표시하였다. 제 1의 Step에서는 SAR 生 Data의 入力로부터 Range 壓縮 결과의 磁氣 Disc에 대한 出力까지를 하지 않고, 제2의 Step에서 磁氣 Disc로부터의 해독을 통해 Azimuth 壓縮 결과인 畫像 Data 出力까지를 실행한다.

Step 1에서 磁氣 Disc에 기록된 Data를, Step 2에서 解讀하면서 實行하는 것으로 Corner Turning을 위해서 處理 Step을 설치할 필요가 없다.

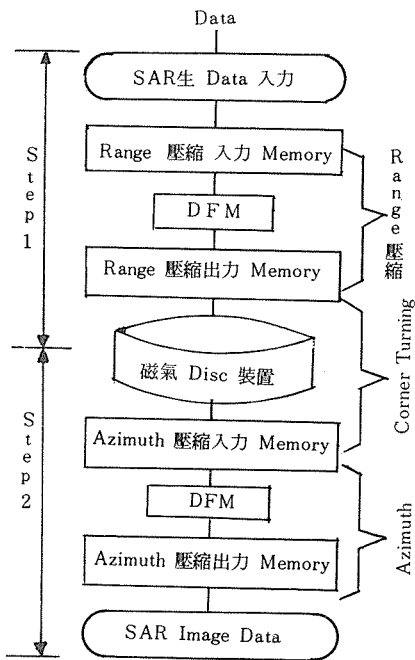


圖 5 NEDIPS에서의 SAR Data 처리의 흐름

磁氣 Disc의 사용시에는, Access에 요하는 시간의 최소로 되게 되어 磁氣 Disc의 Addressing을 연구하고 있다. 表 1은 Range 壓縮·Azimuth 압축에 있어서의 Pipe Line 處理의 형태를 나타낸 것이다. 제각기의 압축시에는 1~4의 처리가 동시에 행하여진다.

NEDIPS를 이용해서, 美國의 海洋觀測衛星 Seasat에 의한 SAR Data(100×50km 상당)를 처리한 경우 약 2시간이 소요되어 할 수 있었다.

表 1 NEDIPS에 있어서의 Pipe Line 動作

Range 壓縮	1. MT → MS Main Memory
	2. MS Main Memory → 外部 記憶 Memory
	3. 外部 記憶 Memory : DFM (Data 入力) DFM (演算結果出力) → 外部 記憶 Memory
	4. 外部 記憶 Memory → 磁氣 Disc 裝置
Azimuth 壓縮	1. 磁氣 Disc 裝置 → 外部 記憶 Memory
	2. 外部 記憶 Memory : DFM (Data 入力) DFM (演算結果出力) → 外部 記憶 Memory
	3. 外部 記憶 Memory → MS Main Memory
	4. MS Main Memory → MT

이 같은 畫像處理에 한정되지 않고, 광범위한 일반 科學技術計算 및 각종 Simulation에 대해서도 제각기의 목적에 맞춘 Hardware System을 이 DFM을 사용하여 校正하는 것에 의해, 종래의 비슷한 형태의 시스템과 비교해 볼 경우 그 성능이 비약적으로 향상된 것임을 알 수 있고, Minicom 정도의 Hardware 규모로써 Super Computer를 훨씬 능가하는 Cost Performance를 實現한 것임을 알 수 있다.

日電의 발표에 의해 세계 각국의 注目을 받고 있는 이 세계 최초의 非 Neumann型 超高速 컴퓨터는, 현재 出荷中인 大型 컴퓨터에 비해 50~100배의 超高速 計算을 실현한 것으로 각광을 받고 있지만, 더욱 큰 관심의 대상이 되고 있는 것은, 이 컴퓨터의 登場으로 하여 理想型으로만 생각해 오던 人工知能의 第5世代 컴퓨터 開發에 획기적인 突破口와 Key 역할을 하게 되었다는 점에 그 核心이 있는 것이다.