

TM-0857,0858
論理型言語処理マイクロプロセッサ

A40-bit Microprocessor for
Logic Programming Language

安藤秀樹, 町田浩久,
中島 浩, 中屋雅夫 (三菱電機)

February, 1990

©1990, ICOT

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191-5
Telex ICOT J32964

Institute for New Generation Computer Technology

論理型言語処理マイクロプロセッサ

A 40-bit Microprocessor for Logic Programming Language

安藤 秀樹
Hidetaki ANDO

町田 浩久 中島 浩
Hirohisa MACHIDA Hiroshi NAKASHIMA
三菱電機株式会社
Mitsubishi Electric Corporation

中屋 雅夫
Masao NAKAYA

1. はじめに

我々は、第5世代コンピュータ・プロジェクトの一環として並列推論マシンPIM/m(Parallel Inference Machine)を開発している。PIM/mは、要素プロセッサを2次元メッシュ状に接続したマシンであり、Prologライクな並列論理型言語を処理する。図1に示すように、要素プロセッサはPU(Processing Unit)、CU(Cache Unit)などからなる。PUは、32KwのWCSに格納されたマイクロプログラムの制御により動作するマイクロプロセッサである。以下、PUのアーキテクチャの特徴及びレイアウト設計について述べる。

2. アーキテクチャ

PUは図2に示すように、命令デコード(D)、オペランド・アドレス計算(A)、オペランド読み出し(R)、マイクロプログラム・ディスパッチ(S)及び実行(E)の5ステージよりなる。

PUでは、論理型言語の処理に特有なタグの判定操作をハードウェア化している。Eステージにおけるマイクロプログラムの制御機能としてのタグによる分岐のほか、A、R、Sステージにはデレフェレンス機能、Sステージにはオペランドのタグによるマイクロプログラムのディスパッチ機能があり、命令実行に先立ってタグ判定を行いEステージにおけるタグ判定のオーバーヘッドを緩和している。

3. レイアウト設計

図3にチップ・プロットを示す。チップは自動配置配線プログラムによってレイアウトされ、13Kのスタンダード・セルとモジュール・ジェネレークにより生成される8つのマクロ・セルよりなる。

また、クロック・スキューを最小にするため図4に示すように、クロック・バッファをチップの両サイドに配置した。バッファからフリップ・フロップまでのクロック線の抵抗を下げるため、垂直チャネルの線幅を広くし、そして、水平なチャネル内で両バッファの出力を結合した。

4. おわりに

表1にチップ諸元を示す。チップは16.7MHzのクロックで動作し、appendで従来[1]の3倍の速度1.28MLIPSを達成することがシミュレーションによって確認されている。

- [1] H. Nakashima et. al, "Hardware architecture of the sequential inference machine: PSI-11", Proc. of the Symp. on Logic Programming, pp.104-113, Aug. 1987.

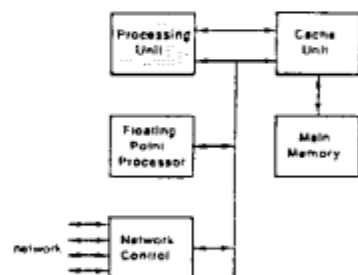


図1 PIMの要素プロセッサ

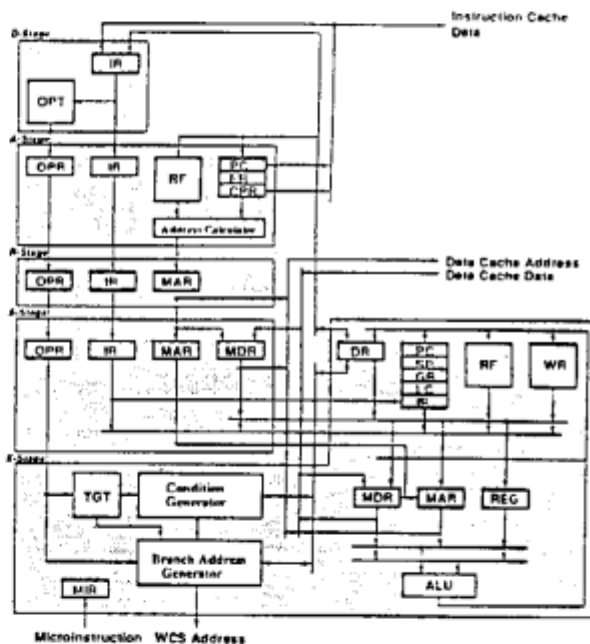


図2 アーキテクチャ

表1 チップ諸元

Data width	40bit
ALU	32bit
Operation decode table	54bit x 512word
Register file	40bit x 32word x 2
Work register	40bit x 32word
Tag dispatch table	8bit x 512word
Microprogram address stack	15bit x 32word
Transistor count	384K
Chip size	16.3 x 13.6 mm
Package	361 pin PGA
Clock	16.7MHz
Power supply	5V
Process technology	0.8um double-metal CMOS

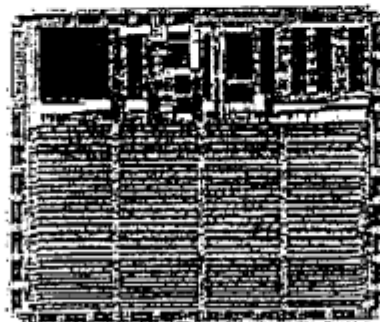


図3 チップ・プロット

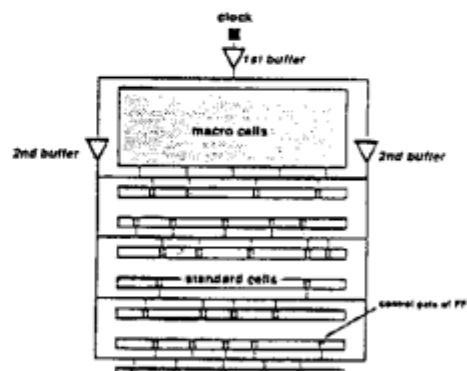


図4 クロック分配