

フィード・フォワード・計算機の性質について

EXPLOITING THE CHARACTERISTICS OF THE FEED-FORWARD-MACHINE

金井 敦

Atsushi Kanai

中村 維男

Tadao M. Nakamura

重井 芳治

Yoshiharu Shigei

東北大学 工学部

Faculty of Engineering, Tohoku University

1. ま え が き

近年、計算速度の向上を目指して、データフロー・マシン等の従来のノイマン型計算機とは異なるアーキテクチャの計算機が提案されている⁽¹⁾。その中で、フィード・フォワード・計算機(FFM)が提案された⁽²⁾。

FFMは、従来のデジタル電子計算機とはその動作概念が異なる計算機である。従来の計算機は、その構造上、命令のフェッチとデコードを行ってからオペランドのフェッチと演算の実行を行っている。これに対して、概念的には、FFMは命令のフェッチとデコードそしてオペランドのフェッチと演算の実行を同時に行う。従来の計算機では、ALUに命令のデコード結果を与えて演算の選択を行っていた。FFMで

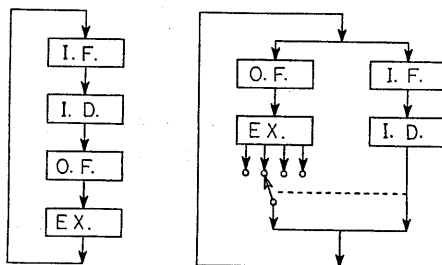
は機能の固定された演算装置を数種並べることにより、命令のデコード結果を待たずにすべての演算を同時に行う。

このような構造を持つFFMでは、効率よく動作させるために、従来とは異なるソフトウェアが必要となると思われる。また、FFMでは多くの演算装置が同時に動作しているために並列処理もある程度可能となる。

本稿では、このFFMにおいて将来の設計指針となりうる演算装置の利用効率ならびにコストを考慮した演算装置の最適な個数について解析的に検討する。

2. FFMの構成

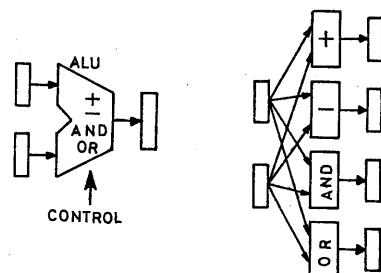
FFMの概念的な動作を図1に示す。同図(a)



(a) 従来方式

(b) FFMの概念

図1 FFMの動作概念



(a) 従来の計算機

(b) FFM

図2 FFMと従来の計算機との対応

に示してあるような従来の計算機の機構語の実行に比べると、同図(b)に示すFFMの動作概念では命令のフェッチおよびデコードと同時にオペランドのフェッチと実行を行っている。

例えば、図2に示すように、ALUの機能として、 $+$ 、 $-$ 、AND、ORを有する従来の計算機に対応したFFMは、独立に $+$ 、 $-$ 、AND、ORを行う演算装置を並べて構成される。このように、FFMではすべての演算を同時に実行するために、命令のデコードを待つことなく演算の実行が可能となる。

図3にFFMの演算実行部の基本的構成を示す。同図で、MUXはコントロール信号により演算結果を選択するマルチプレクサである。選択されたデータは新たにオペランド入力として加えられる。

図4に具体的なFFMの演算実行部の構成を示す。ここでは、演算装置(PE)にはそれぞれバッファが設けられる。このバッファは、演算装置から見るとスタックのように見え、出力結果は次々とバッファに積まれ、バッファを溢れた結果はバッファの右端から消滅するものとする。計算機システムとしての性質上、この演算実行部の外部に演算データや初期値を入れるメモリが用意してあるが、FFMの動作の基本はバッファからオペランド・レジスタへのデータ転送である。

FFMの全体の構成を図5に示す。ここでは、プログラム・メモリ、データ・メモリ、演算実行部およびコントロール部がある。プログラム・

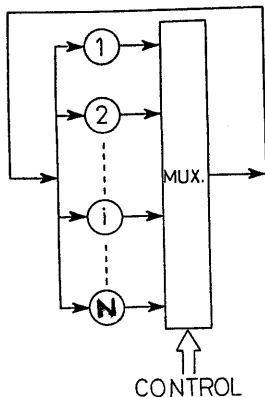


図3 実行部の基本的構成

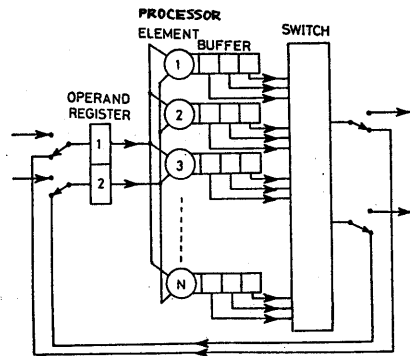


図4 実行部の具体的構成

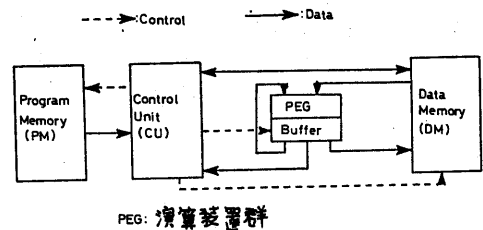


図5 FFMの全体的構成

メモリにはプログラムが、データメモリには演算に使用されるデータおよび演算結果が格納される。

3. FFMの特性

FFMは命令のデコードを待たずに演算を実行できるので、従来の計算機が命令のデコードを行ってから演算を行っていたのに比較して高速化が期待できる。また、多数の処理装置により並列処理も可能となり、このレベルでも高速化が期待できる。

例えば、演算装置として $+$ 、 $-$ 、 $*$ 、 $/$ を持ったFFMを考える。そして、複素数 $(a+bi)$ と $(c+di)$ の和、差、積、商を求める場合を

$$(a+bi)/(c+di) = \frac{(ac+bd)+(bc-ad)i}{c^2+d^2}$$

if $a+b=\emptyset$ then a/b
else $a*b$ (1)

FFMの稼働時間(内部処理時間のみ考える)を T_p とする。稼働時間に対して、条件分岐等

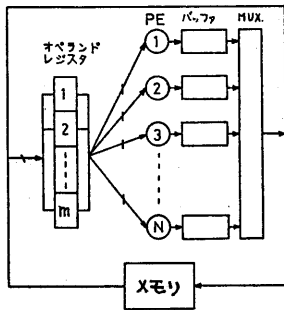


図7 FFMのモデル

でオペランド・レジスタへのデータ転送が中断し、データを連続してセットできない場合の割合を δ とする。

図8は、FFMの動作のようすを示したものである。図8の $a_1, a_2, \dots$ でオペランド・レジスタにデータがセットされる。次に、演算装置の実行が始まり、同時に、命令がデコードされて、オペランド・レジスタへのデータ転送が行われる。 a_1, a_2, a_3 では、 T_B / T_M のサイクルに対し連続してデータがオペランド・レジスタにセットされているが、 a_4 では a_3 で実行された結果が出るまで次のステップへは進めない場合（条件分岐）を示している。したがって、本来 T_B が T_M 周期で進行する処理がそれより長い時間を必要とする。

図9に、オペランド・レジスタへのデータセットのようすを示す。縦の m 個がそれぞれのオペランド・レジスタに対応し、横が図8の時刻 $a_1, a_2, \dots$ におけるオペランドの格納のようすを示している。

稼動時間中にオペランド・データが到着する回数は、

$$\frac{T_P \cdot (1 - \delta)}{\bar{T}} \quad (2)$$

で与えられる。ここで、 $\bar{T}$ は T_P 内で T_B の出現回数を n_1 、 T_M の出現回数を n_2 とすると、

$$\begin{aligned} \bar{T} &= \frac{n_1 \cdot T_B + n_2 \cdot T_M}{n_1 + n_2} \\ &= \theta \cdot T_B + (1 - \theta) T_M \\ \theta &= \frac{n_1}{n_1 + n_2} \end{aligned} \quad (3)$$

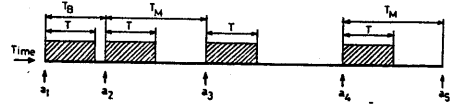
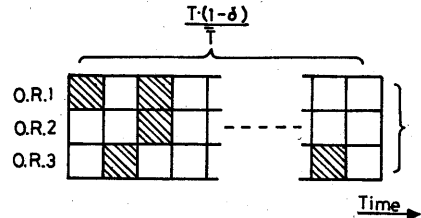


図8 FFMの動作のようす



O.R. : オペランド・レジスタ

斜線 : メモリから転送されたデータ

図9 オペランド・レジスタへのデータセットのようす

で与えられる。これは、サイクルの平均時間である。従って、使用された全オペランド・データ数 M_T は次式となる。

$$M_T = m \frac{T_P \cdot (1 - \delta)}{\bar{T}} \quad (4)$$

次に、演算装置からバッファに出力されるデータについて考える。バッファにセットされるデータのようすを図10に示す。縦は各プロセッサに対応し、横は各プロセッサが出力したデータの時間変化である。

FFMの基本的動作として、バッファからオペランド・レジスタへのデータ転送が考えられるが、バッファからオペランド・レジスタへの転送はバッファに出力された1個のデータが一度だけオペランド・レジスタへ転送されるとは限らない。同じデータが2度、3度とオペランド・レジスタへ転送される可能性がある。図9のように、オペランド・レジスタに注目して、メモリから転送されたデータ（図中斜線）を除いた部分を考える。残りのデータはすべてバッファから転送されたものである。バッファから転送されたデータのなかで、前に述べたように、オペランド・レジスタ側から見ると複数のデータに見えても（論理的）バッファからは1個のデータが転送される場合（物理的）もある。

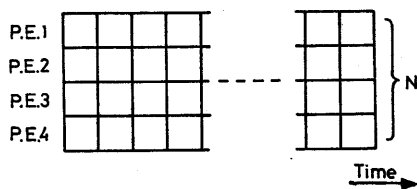


図10 バッファへのデータセットのようす

そこで、図10において、バッファから転送されたオペランド・データの中で、バッファから同一データが1度だけオペランド・レジスタに転送されたデータの割合を p_1 、2度転送されたデータの割合を p_2 、以下 n 回転送されたデータの割合を p_n とする。

$$\sum_{i=1}^{\infty} p_i = 1 \quad (5)$$

また、図9の斜線を除いた残りのデータの個数は、

$$M_T - D_I \quad (6)$$

となる。ここで、 D_I はメモリからオペランド・レジスタへ転送されたデータの全個数である。したがって、例えば、バッファから同じデータが2度オペランド・レジスタに転送されたデータの場合、オペランド・レジスタ側ではそのデータの個数は、 $(M_T - D_I) \cdot p_2$ となるが、バッファでの実際のデータの個数は、

$$\frac{(M_T - D_I) \cdot p_2}{2} \quad (7)$$

となる。このようにして、バッファ側から見ても、オペランド・レジスタへ転送された全データ M_R は次式で示される。

$$M_R = (M_T - D_I) \cdot \sum_{i=1}^{\infty} \frac{p_i}{i} \\ = \left\{ \frac{m \cdot T_P \cdot (1 - \delta)}{T} - D_I \right\} \cdot \sum_{i=1}^{\infty} \frac{p_i}{i} \quad (8)$$

ところで、演算装置が出力したデータの中で、結果的に有効となるものは、オペランド・レジスタに転送されたデータと出力結果として有効となるデータである。そこで、オペランド・レジスタへ転送されはしなかったが、出力される結果として有効となるデータの個数を D_o とすると、演算装置が出力したデータの中で有効と

なった全データ数 D_e は次式で与えられる。

$$D_e = M_R + D_o \\ = \left\{ \frac{m \cdot T_P \cdot (1 - \delta)}{T} - D_I \right\} \cdot \sum_{i=1}^{\infty} \frac{p_i}{i} + D_o \quad (9)$$

4.1 FFMのコストとスルー・プット

FFMのスルー・プット H を、単位時間当りに有効となる演算数とする。稼働時間 T_P 内で有効となった全演算数は D_e であるから、FFMのスルー・プット H は式(9)より、

$$H = \frac{D_e}{T_P} \\ = \frac{1 - \delta}{T} \cdot \left\{ (m - \beta_I) \sum_{i=1}^{\infty} \frac{p_i}{i} + \beta_o \right\} \quad (10)$$

となる。ここで、 β_I 、 β_o はそれぞれ、

$$\beta_I = \frac{D_I}{\frac{M_T}{m}} \quad (11)$$

$$\beta_o = \frac{D_o}{\frac{M_T}{m}} \quad (12)$$

であり、前者は D_I の平均個数を表わしている。ただし、

$$0 \leq \beta_I \leq m \quad (13)$$

である。また、後者は D_o の平均個数を表わしている。ただし、

$$0 \leq \beta_o \leq N \quad (14)$$

である。

FFMを構築するためのコスト C_{FFM} を考える。素子の性能に逆比例した時間である T_B を評価基準とする。いま、 T_B を固定しく(T_B^o)それを実現するためにかかるコストを K とすると任意の T_B のFFMのコスト C_{FFM} は次式で与えられる⁽³⁾。

$$C_{FFM} = T_B^{\alpha} \cdot K \cdot \left(\frac{1}{T_B} \right)^{\alpha} \quad (15)$$

ここで、 α は0.5～1程度の定数である。式(15)で T_B を用いたが、 T_B / T_M は一定と考えられるので、 T_B のかわりに T_M を用いても良い。

また、ここでは、 T/T_B も一定とする。

コスト K は、演算装置個数 N に比例する部分と N に依存しない部分（制御装置等）がある。したがって、 K は次式で表わされる。

$$K = C_0 + C_p N$$

$$= C_p \cdot (A_F + N) \quad (16)$$

式(15)より次式が得られる。

$$C_{FFM} = T_B^{\alpha} \cdot C_p \cdot (A_F + N) \cdot \left(\frac{1}{T_B}\right)^{\alpha}$$

$$= C_1 \cdot (A_F + N) \cdot \left(\frac{1}{T_B}\right)^{\alpha} \quad (17)$$

ここで、 $C_1 = T_B^{\alpha} \cdot C_p$ である。

次に、 $\bar{T}$ を $T_M = k \cdot T_B$ として T_B で表わすと、式(3)より次式となる。

$$\bar{T} = \delta \cdot T_B + (1 - \delta) \cdot k \cdot T_B$$

$$= Q \cdot T_B \quad (18)$$

式(18)を式(10)へ代入して T_B について解くと次式となる。

$$T_B = \frac{1 - \delta}{Q \cdot H} \left\{ (m - \beta_1) \sum_{l=1}^{\infty} \frac{p_l}{l} + \beta_0 \right\} \quad (19)$$

式(19)を式(17)に代入すると、

$$C_{FFM} = C_1 (A_F + N) \left\{ \frac{Q \cdot H}{(1 - \delta) \cdot \left\{ (m - \beta_1) \sum_{l=1}^{\infty} \frac{p_l}{l} + \beta_0 \right\}} \right\}^{\alpha}$$

$$(20)$$

となる。

4.2 演算装置の利用効率

演算装置1個当たりで有効となった全データ数は、 D_e/N で与えられる。演算装置1個が一つのデータを生成するのに時間 T だけ必要であるから、このデータを生成するのにかかった時間は、

$$\frac{D_e \cdot T}{N} \quad (21)$$

である。したがって、稼働時間に対する演算装置の有効な動作時間の割合をその利用率 η_p と定義すると、 η_p は次式で与えられる。

$$\eta_p = \frac{D_e \cdot T}{N \cdot T_A}$$

$$= \frac{T(1 - \delta)}{N \cdot \bar{T}} \left\{ (m - \beta_1) \sum_{l=1}^{\infty} \frac{p_l}{l} + \beta_0 \right\} \quad (22)$$

4.3 数値例と検討

式(20)と式(22)を計算するために、バッファからオペランド・レジスタへ重複してデータが転送される割合 p_k の分布は、実際に測定しなければ求まらない。そこで、ここでは、 p_k の分布をポアソン分布と仮定して計算を行う。一般的に考えて、演算装置個数 N が多くなるとバッファからオペランド・レジスタへデータが転送されるときに、同じデータが何度も転送される率は統計的に低下すると思われる。以上を考慮して、 p_k を次式のように表わす。

$$p_k = e^{-\lambda} \frac{\lambda^{k-1}}{k-1} \quad (23)$$

ここで、 λ を、

$$\lambda = C \cdot \frac{1}{N} \quad (C: \text{定数}) \quad (24)$$

とする。すると、

$$\sum_{l=1}^{\infty} \frac{p_l}{l} = \frac{1 - e^{-\lambda}}{\lambda} \quad (25)$$

となる。

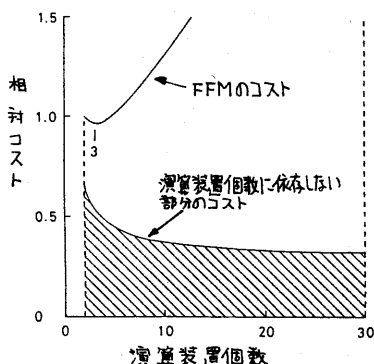
次に、 A_F を求めるために、現在設計中の FFM の A_F を求める。この FFM の構成を以下に述べる。演算装置として TTL の ALU (16 ビット) を用い、バッファにアクセス・タイム 70 ns の C-MOS (演算装置1個当たり、1 K X 36 bits) を用いる。また、コントロール部には Am 2903 X 4 を用い、データメモリは 500 ns の N-MOS (2 K X 16 bits)、そしてプログラムメモリは同じく 500 ns の N-MOS (16 K X 16 bits) を用いる。この場合、IC のコストのみ考慮すると、ほぼ $A_F = 4$ となる。しかし、FFM 専用の IC を使用すれば、 A_F はより増大するものと思われる。

さて、コスト計算において、 A_F に関しては $A_F = 4$ 、10 という値を用いてみた。α に関しては、現在まで α は 0.5 と考えられていたが、

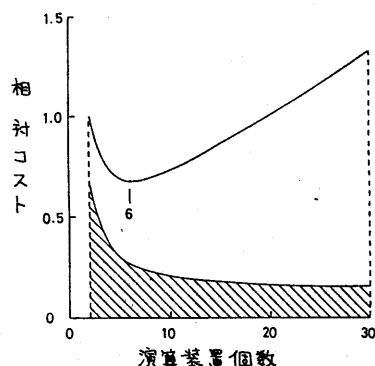
今後は、増大する傾向があると思われる。そこで、 $\alpha = 0.5, 1$ について計算する。また、 T_B と T_M については設計値より、 $T_B = 600\text{ns}$ 、 $T_M = 1000\text{ns}$ を用い、その他のパラメータについては、 $\delta = 0.1$ 、 $\gamma = 0.5$ 、 $\beta_i = \beta_o = 0$ 、 $m = 2$ 、 $C = 10$ 、 $H = 2 \times 10^6$ で計算を行った。図11に一定スループットの時に、 N を変化させたときのコスト C_{FFM} のようすを示す。実用上、 $m < N$ であるので、同図では N は2から始まっている。このグラフは、 $N = 2$ のときのコストを1とした相対値である。演算装置個数に依存しない部分のコストは、式(20)中 N を除いたものから導出された。同図より、 A_F が大きくなると α が大きいほうが、演算装置の最適個数が多くなる

ことがわかる。

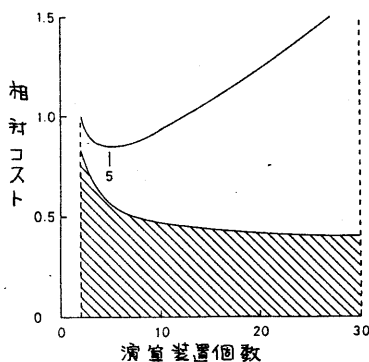
次に、演算装置の利用率について、 $T = \pi$ 、 $\delta = 0.1$ 、 $\beta_i = \beta_o = 0$ 、 $C = 8$ 、 $m = 2$ としたときのように図12に示す。同図より、演算装置個数 N が多くなると演算装置の利用率が減少することがわかる。しかし、演算装置の機能がALUレベルのFFMの場合、普通の計算機と比較して利用率がそれほど悪いとは思われぬ。普通の計算機のALUでも演算の種類毎にハードウェアが分離していれば、同時に複数の演算は行えないことがむしろハードウェアの無駄が、ALUよりも少ないとも考えられる。しかし、演算装置としてプロセッサ等を用いるFFMでは利用率が問題となってくると思われる。図12



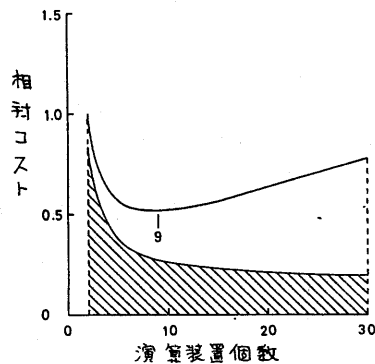
(a) $\alpha = 0.5, A_F = 4$



(b) $\alpha = 1, A_F = 4$



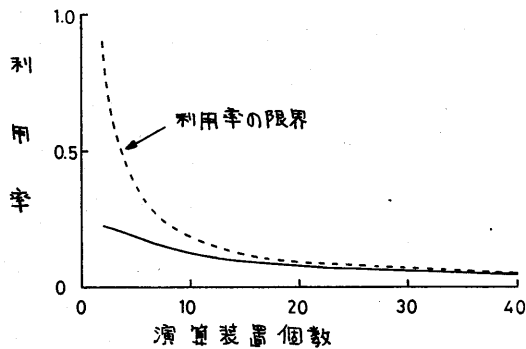
(c) $\alpha = 0.5, A_F = 10$



(d) $\alpha = 1, A_F = 10$

$C = 10, m = 2, H = 2 \times 10^6, \gamma = 0.5, \delta = 0.1, \beta_i = \beta_o = 0, T_B = 600\text{ns}, T_M = 1000\text{ns}$

図11 演算装置個数とコストの関係



$m=2, T=\bar{T}, \delta=0.1, \beta_E=\beta_0=\emptyset, C=8$

図12 演算装置個数と利用率の関係

では、 $\beta_0=\emptyset$ として計算したが、実際の問題では β_0 がかなり大きくなる分野があると思われる。したがって、図12はFFMの一悪い状態の利用率を見ていることになる。

5. ま と め

コストを考慮した演算装置の最適個数について検討した。その結果、演算装置個数が数個の近辺に最適個数が存在することがわかった。その分布については、演算装置の機能やプログラム等によっても様々に変化するものと思われる。しかし、この分布は解析的に求めるのは非常に難しく、経験により求める必要があると思われる。

また、利用率については、演算装置が多くなるほど利用率が悪くなることがわかった。しかし、これはALLレベルのFFMではその演算装置個数を限定すれば、ほとんど問題にならないと思われる。

今後は、FFMをより有効に動作させるソフトウェアについて検討する必要がある。

参 考 文 献

- (1) Treleaven, P.C.: "Exploiting Program Concurrency in Computing System", IEEE C.M. pp.42-50 (Jan. 1979).
- (2) 重井, 長谷川, 中村: "Feed Forward 計算機の提案", 昭和55, 情報処理学会第21回全国大会, 2J-5.
- (3) 猪瀬, 菅島: "電子計算機的方式設計", 産報出版, 昭和53.
- (4) 元岡: "計算機システム技術", オーム社, 昭和53.
- (5) 金井, 長谷川, 中村, 重井: "フィード・フォワード・計算機の構造", 信学技報 E C 80-34 (1980-09).
- (6) 金井, 長谷川, 中村, 重井: "フィード・フォワード・計算機におけるプログラミングの考察", 昭和56, 信全大.
- (7) 金井, 中村, 重井: "フィード・フォワード・計算機における演算装置の利用率に関する検討", 昭和56, 電気関係学会東北支部連合大会, 1H7.