単一プロセッサシステムにおけるキャッシュ，ストアバッファとメモリインタリーブの効果と多重プロセッサシステムの性能について†

齋藤 将人††

ハードウェアテクノロジ，ハードウェア論理構造，ソフトウェア構造とシステムアーキテクチャの特性を表わす若干のパラメータを使用して，ミスヒット率 α のキャッシュと各メモリバンク対応に s 段のストアバッファをもつプロセッサと n バンクのインタリーブされたメモリユニットで構成される単一プロセッサシステム $((\alpha, s, n))$ の平均命令実行時間の解析を，$(\alpha, 0, 1)$, $(\alpha, 1, n)$, $(\alpha, s, 1)$ の各ケースについて行った．プログラムの実行過程を演算サイクルとメモリ要求の交互動作ととらえると，ストアバッファは演算サイクル時間とメモリサイクル時間がバランスしている領域で効果が大きく，キャッシュとメモリインタリーブはメモリサイクル時間が演算サイクル時間に比べ大きくなるにつれ効果が大きくなり，一般に実際に走行するメモリサイクルの回数を減少させることが性能向上に寄与する．

更に，各プロセッサの単一プロセッサシステムでのメモリ使用率がわかっているとき，任意数のプロセッサからなるインタリーブされたメモリにより密結合された多重プロセッサシステムのシステム性能と各プロセッサの平均命令実行時間を解析した．

この単一と多重の両プロセッサシステムの解析値例は，実際の大型機の論理構造をモデルにしたシミュレーション結果に対して +1～3% の差異であった．

1. まえがき

筆者は，プロセッサの動作を演算サイクルとメモリサイクルの交互動作としてとらえ，ソフトウェア，システムアーキテクチャ，ハードウェア論理構造とテクノロジの各特性を表わす若干のパラメータを使用してキャッシュもストアバッファもない単一プロセッサの性能解析を行い[1]，この拡張として1メモリユニットを共用する2プロセッサシステムの解析[2]と，更にこれの n メモリユニットへの拡張を行った[3]．TAKÁCS[4]が解析した特性が同じ m 個の automatic machines と1 repaireman の待時間は筆者の特殊ケースと一致した．Skinner ら[5]はプロセッサが1つのメモリユニットを指定したときそれが閉塞されている確率を一定値としてプロセッサとメモリユニットが各任意数のシステムのメモリ競合問題の解法を提案したが，これはメモリ使用率が小さいと筆者の結果と数値的にほとんど差異がなく，メモリ使用率が大きいと差異が出てくるが高々5%程度である[3]．

ここでは，更に，キャッシュとストアバッファとメモリインタリーブを導入し，これらの具備の有無でできる典型的な5つのタイプの単一プロセッサシステムの性能解析と，各プロセッサの単一プロセッサシステムでのメモリ使用率がわかっているとき，インタリーブされたメモリをもつ任意数の多重プロセッサシステムの性能を解析し，この理論と実際の装置をモデルにしたシミュレーション結果との比較を行う．

2. 単一プロセッサシステムの解析

2.1 解析するモデルと基本動作

演算ユニットを主体に付加的にキャッシュとストアバッファをもつプロセッサとメモリユニットからなるシステムを考える．

演算ユニットは命令読出の起動，命令の解釈，オペランドのメモリ番地生成，オペランドデータ読出の起動，演算と結果のメモリユニットへの書込（ストア）の起動等を行う．メモリ読出または書込の起動の間に行う演算を演算サイクルと呼ぶ．

仮定(1) 任意の演算サイクルの時間 u は次の確率分布関数 $f(u)$ をもつ．ただし，λ は定数で演算サイクルのパラメータと呼ぶ．

$$f(u)=\lambda \exp(-\lambda u). \qquad (1)$$

演算サイクルは読出または書込のメモリ要求で終了するがその確率を β_R または β_S とする．

$$\beta_R+\beta_S=1 \qquad (2)$$

メモリユニット（単にメモリと呼ぶ）は命令とデータ（両者を合わせて情報と呼ぶ）を格納しており演算ユニットの要請に応じて情報の読出，書込の動作を行

† Evaluation of Cache Memory, Store Buffer and Memory Interleaving on Single-processor System, and Its Application to Multiprocessor System Environment by MASATO SAITO (Computer Engineering Division, Nippon Electric Co., Ltd.).

†† 日本電気(株)コンピュータ技術本部

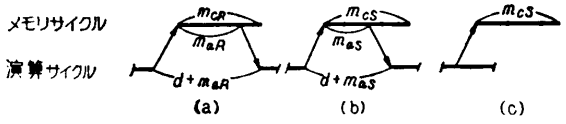

図 1 演算サイクルとメモリサイクル
Fig. 1 The processing cycle and the memory cycle.

う．読出または書込の動作を読出メモリサイクルまたは書込メモリサイクルと呼び両者を合わせて単にメモリサイクルと呼ぶ．後述するキャッシュへの転送を除き一回のメモリ要求に対しては1メモリサイクルが走行する．読出と書込のサイクル時間を各々 m_{cR} と m_{cS} とし，読出アクセス時間を m_{aR} とする．書込メモリサイクルでは次のメモリ要求が受付可能になった時点でメモリから応答信号を返すものとし，これを書込応答時間と呼び m_{aS} とする．キャッシュまたはストアバッファとメモリ間の論理遅延時間を片道 $d/2$ とする．したがって，実効読出アクセス時間と実効書込応答時間は**図1**(a)(b)のように $d+m_{aR}$，$d+m_{aS}$ となる．インタリーブされたメモリではインタリーブ数だけメモリサイクルを並列に走らすことができるが，この動作が独立したハードウェアの実体をバンクと呼び，このバンク数を n で表わす．

仮定(2) 任意のメモリ要求でのバンク指定はランダムである．

キャッシュはメモリのバッファであり，情報の読出要求が出るとまずキャッシュ内に該当情報があるかどうか調べ，あれば（ヒット）これを，なければ（ミスヒット）メモリまたは後述のストアバッファから取出したものを演算ユニットに与える．キャッシュの動作をキャッシュサイクルと呼び，キャッシュ内の情報をキャッシュのエントリと呼ぶ．任意の読出メモリ要求がキャッシュでミスヒットする確率を α とする．キャッシュがあるシステムでは1読出メモリ要求に対してキャッシュの1ブロック分の情報をメモリから移送するため複数回（ζ とする）の転送を行うことがある．

仮定(3) キャッシュのあるシステムでは1読出メモリ要求に対し ζ 回の読出メモリサイクルを実行しその最初の情報到着時点で次の演算サイクルが始動する．

複数回転送の最後の情報到着を待って演算サイクルが始動する場合は以下の議論で，μ をクロック周期時間として，実効読出アクセス時間を $d+m_{aR}+\mu(\zeta-1)$ とすればよい．

ストアバッファはメモリへの書込情報の一時格納手段で演算ユニットはストアバッファに情報を転送することで書込動作が終了したとみなす．ストアバッファにある情報（エントリ）はメモリの空き時間を使って順次書込みされる．したがって，ストアバッファがあると，書込メモリサイクルと並行して演算サイクルの走行ができ（図1(c)）演算ユニットから書込応答時間は見えない．ここでは，書込メモリサイクル中はストアバッファの1段が占有されるものと考える．

仮定(4) 読出メモリ要求は当該バンクの実行中メモリサイクル終了後ほかの書込要求に優先して処理される．

各メモリバンク対応に2段以上のストアバッファをもつシステムでは，キャッシュでミスヒットしてもストアバッファに該当エントリがあればそれを演算ユニットに与え読出メモリサイクルは行わない．キャッシュでミスヒットしたことがわかっているときストアバッファでもミスヒットする確率を α_S とする．また，ストアバッファの動作をストアバッファサイクルと呼ぶ．

仮定(5) キャッシュとストアバッファサイクルのいずれも演算サイクルに繰入れを扱うかまたは演算サイクル時間に比べ小さいと考えるかして0時間で扱う．

仮定(6) $m_{cR}, m_{cS}, m_{aR}, m_{aS}, d, \alpha, \alpha_S$ は常に一定値とする．

メモリ要求が出ると演算サイクルは終了するが，読出メモリ要求でキャッシュかストアバッファかでヒットすると演算サイクルが継続してみえる．この重畳された演算サイクルをマクロな演算サイクルと呼び，この時間が U である確率を $F(U)$ とする．明らかに，

$$F(U)=\lambda_M \exp(-\lambda_M U), \quad (3)$$

$$\lambda_M=(\beta_R\alpha\alpha_S+\beta_S)\lambda, \quad (4)$$

または，$=(\beta_R\alpha+\beta_S)\lambda$ （$\alpha_S=1$ のとき） (5)

で，λ_M をマクロな演算サイクルのパラメータと呼ぶ．

$m_{cR}, m_{cS}, d+m_{aR}, d+m_{aS}$ の大小関係で種々の解析が可能であるが，ここでは次の仮定を設ける．したがって，演算サイクルが終了したときメモリサイクル中であればそれは書込メモリサイクルである．

仮定(7) $d+m_{aR}\geq m_{cR}$，$d+m_{aS}\leq m_{cS}$. (6)

メモリ待は演算サイクルまたは（キャッシュまたはストアバッファがあるとき）マクロな演算サイクル終了時点で指定したメモリバンクが書込メモリサイクル中のとき生ずるが，ストアバッファのあるシステムでは，書込要求が出てもメモリバンクに対応したストア

バッファが満杯になるまでメモリ待を生じない．今，演算サイクルまたはマクロな演算サイクルの終了直前でアクセスしようとするメモリバンクが書込サイクル中でない確率を φ_0 とし，書込サイクル中でその残り時間が t より大きい確率を各メモリバンク対応のストアバッファが1段以下のシステムでは $\varphi(t)$，$s+1$ 段のシステムでストアバッファのエントリ数が $s+1$ のとき $\varphi(s,t)$ とする．

$\varphi(m_cS-d-m_as)=0$：ストアバッファなし，(7)

$\varphi(m_cS)=0$：各バンク対応にストアバッファ1段のタイプ (8)

$\varphi(s,m_cS)=0,\ (s=0,1,\cdots,s)$ (9)

$\varphi_0+\varphi(0)=1$ (10)

$\varphi_0+\varphi(0,0)+\varphi(1,0)+\cdots+\varphi(S,0)=1$ (11)

また，メモリ待になることがわかっているときの平均メモリ待時間 $\tilde{\omega},\tilde{\omega}(s)$ は次式となる．

$$\tilde{\omega}=-\frac{1}{\varphi(0)}\int_0^{m_cs-d-m_as} t\frac{d\varphi(t)}{dt}dt:$$ ストアバッファなし (12)

$$\tilde{\omega}=-\frac{1}{\varphi(0)}\int_0^{m_cs} t\frac{d\varphi(t)}{dt}dt:$$ 各バンク対応にストアバッファ1段のタイプ (13)

$$\tilde{\omega}(s)=-\frac{1}{\varphi(s,o)}\int_0^{m_cs} t\frac{d\varphi(s,t)}{dt}dt \quad (14)$$

2.2 プロセッサ性能とメモリ使用率

プログラムの実行過程（問題の処理過程と呼ぶ）は演算サイクルまたはマクロな演算サイクルとメモリサイクルとの交互動作としてとらえることができるが，2つの連続した演算サイクルの開始時点間を演算サイクル当りの処理時間と呼びその平均時間を $\tilde{\tau}$ とし，命令当りの平均メモリ要求回数を $\tilde{\gamma}$ とすると平均命令実行時間 $\tilde{t}$ は次式となる．

$\tilde{t}=\tilde{\gamma}\tilde{\tau}$ (15)

単位時間当りのメモリの延動作時間をメモリバンク数で割った値をメモリ使用率と呼び l で表わす．今，

$m=\beta_R m_{cR}+\beta_S m_{cS}$ (16)

としたときメモリサイクル時間 (m) と平均演算サイクル時間 $(1/\lambda)$ との相対値 l_0 を次式で定義する．

$l_0=m/(m+1/\lambda)$ (17)

2.3 各タイプの性能解析

システムの形をタイプ（キャッシュの有無（有=1，無=0），ストアバッファ段数（0はストアバッファなし），メモリバンク数）の記号で表示する．$\tilde{\tau},\tilde{t},l$ で特にタイプを区別するときはそれらにタイプの記号を添字として付すことにする．

ストアバッファのないタイプでは **2.1** のストアバッファの定義と式(6)の仮定からメモリインタリーブの効果があまり期待できないことからメモリバンク数は1とする．メモリインタリーブのあるタイプでは，メモリバンクが閉塞されていないかぎり次々と異なったバンクに書込メモリサイクルを走行させうるから各メモリバンク対応に1段のストアバッファをもつことにする．バンク当り複数段のストアバッファの効果はメモリバンク数1のタイプで評価する．したがって，ここでは，タイプ $(0,0,1)$，$(1,0,1)$，$(0,n,n)$，$(1,n,n)$，$(1,s+1,1)$ を解析する．

a) タイプ $(1,n,n)$

n バンクのメモリと各バンク対応に1段ストアバッファをもちキャッシュのあるタイプである．

図2 にこのタイプでのメモリ待事象を示した．同図の 1，i'，$i+1$，j はマクロな演算サイクルの終了時点で λ_M は式(5)で与えられる．図2(a)は，時点1でメモリ待にならない書込メモリ要求が発行され（確率 $\beta_s\varphi_0/(\beta_R\alpha+\beta_s)$），その後時点 $i+1$ までの U 時間 $(0\leq U\leq m_cs-t)$ で i 回のマクロな演算サイクルを実行し（確率 $\lambda_M^{i}U^{i-1}\exp(-\lambda_M U)/(i-1)!$）そのうち最後を除いて $i-1$ 回はこのメモリサイクル中でないバンク指定の書込メモリ要求で終了したが（確率 $(\beta_s/n(\beta_R\alpha+\beta_s))^{i-1}(n-1)!/(n-i)!$）最後は同じバンク指定（確率 $1/n$）の書込または読出メモリ要求が発行され（確率 1），この時点から時点 $i+2$ までメモリ待となりその間演算サイクルは停止する．図2(b)は時点1で書込メモリ要求が発行されてメモリ待を生ずる（確率 $\beta_S\varphi(0)/(\beta_R\alpha+\beta_S)$）が，時点 $1'$ 以降は図2(a)の時点1以降と同じ過程をへる．図2(c)は(a)の事象に加えて時点 i' で発行の書込メモリ要求と同じバンクを時点 j で指定した場合で，次の仮定を設ける．

仮定(8) メモリ待事象が生じたあとそれ以前発行のメモリサイクル上でメモリ待になる確率は無視する．

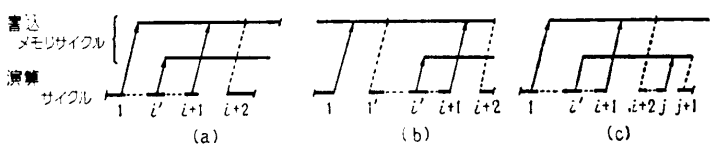

図 2 インタリーブされたメモリユニットでの待事象
Fig. 2 Waiting events on the interleaved memory unit.

仮定(8)から $\varphi(t)$ は図2(a)と(b)の時点 $i+1$ の前記確率事象で形成されるから式(5), (10)を念頭におき, i について1から n まで加えると

$$\varphi(t)=\frac{\lambda\beta_S}{n}\int_0^{m_{cS}-t}\left(1+\frac{\lambda\beta_S U}{n}\right)^{n-1}\exp(-\lambda_M U)dU, \tag{18}$$

また, 平均メモリ待時間は式(13), (18)から次式となる.

$$\tilde{\omega}=\frac{1}{\varphi(0)}\frac{\lambda\beta_S}{n}\int_0^{m_{cS}} t\left(1+\frac{\lambda\beta_S(m_{cS}-t)}{n}\right)^{n-1}\cdot\exp\{-\lambda_M(m_{cS}-t)\}dt. \tag{19}$$

マクロな演算サイクルで平均 $1/(\beta_R\alpha+\beta_S)$ 回の演算サイクルが走るから演算サイクル当りの平均メモリ待時間は $\varphi(0)\tilde{\omega}(\beta_R\alpha+\beta_S)$ となり, このタイプの演算サイクル当りの平均処理時間 $\tilde{\tau}_{(1,n,n)}$ は式(5), (10)を念頭におき式(19)から次式で与えられる.

$$\tilde{\tau}_{(1,n,n)}=1/\lambda+\beta_R\alpha(d+m_{aR})+(\beta_S/n)I(m_{cS},\alpha,n) \tag{20}$$

$$I(m_{cS},\alpha,n)=\lambda_M\int_0^{m_{cS}} t\left(1+\frac{\lambda\beta_S(m_{cS}-t)}{n}\right)^{n-1}\cdot\exp\{-\lambda_M(m_{cS}-t)\}dt^{*} \tag{21}$$

読出メモリ要求当り ζ 回の読出メモリサイクルが行われるとすると, 演算サイクル当りは $\beta_R\alpha\zeta+\beta_S$ 回のメモリサイクルが行われ, メモリは n バンクであるからこのタイプのメモリ使用率は次式となる.

$$l_{(1,n,n)}=(\beta_R\alpha\zeta m_{cR}+\beta_S m_{cS})/n\tilde{\tau}_{(1,n,n)}. \tag{22}$$

b) タイプ $(0,n,n)$

タイプ $(1,n,n)$ でキャッシュをとったものである. 式(20)～(22)で $\alpha=\zeta=1$ とおき, 式(16)を考慮して演算サイクル当りの平均処理時間とメモリ使用率は,

$$\tilde{\tau}_{(0,n,n)}=1/\lambda+\beta_R(d+m_{aR})+(\beta_S/n)I(m_{cR},1,n) \tag{23}$$

$$l_{(0,n,n)}=m/n\tilde{\tau}_{(0,n,n)} \tag{24}$$

となる.

c) タイプ $(1,0,1)$

バンク数1のメモリとキャッシュをもちストアバッファのないシステムである. ストアバッファがないと書込メモリ要求のたびにすくなくとも $d+m_{aS}$ だけおくれて次の演算サイクルが開始される(図1(b)). したがって式(20)に $\beta_S(d+m_{aS})$ を加え, 式(6)を考慮して式(20)の m_{cS} を $m_{cS}-d-m_{aS}$ と置換え, $n=1$ とおくとこのタイプの演算サイクル当りの平均処理時間が得られ, これで式(22)の分母を置換え $n=1$ とすればこのタイプのメモリ使用率が得られる.

$$\tilde{\tau}_{(1,0,1)}=1/\lambda+\beta_R\alpha(d+m_{aR})+\beta_S\{m_{cS}-(1/\lambda_M)(1-\exp(-\lambda_M(m_{cS}-d-m_{aS})))\} \tag{25}$$

$$l_{(1,0,1)}=(\beta_R\alpha\zeta m_{cR}+\beta_S m_{cS})/\tilde{\tau}_{(1,0,1)}. \tag{26}$$

d) タイプ $(0,0,1)$

バンク1のメモリをもちキャッシュもストアバッファもないシステムである. 式(25), (26)で $\alpha=\zeta=1$ とおき式(2), (5), (16)を考慮してこのタイプの演算サイクル当りの平均処理時間とメモリ使用率は次式となる.

$$\tilde{\tau}_{(0,0,1)}=\beta_R(d+m_{aR}+1/\lambda)+\beta_S\{m_{cS}+(1/\lambda)\cdot\exp(-\lambda(m_{cS}-d-m_{aS}))\} \tag{27}$$

$$l_{(0,0,1)}=m/\tilde{\tau}_{(0,0,1)}. \tag{28}$$

今, $m_{cR}=m_{cS}=d+m_{aR}=d+m_{aS}$ (29)

とすると式(27), (28)は式(2), (16), (17)から

$$\tilde{\tau}_{(0,0,1)}=m+1/\lambda \tag{30}$$

$$l_{(0,0,1)}=m/(m+1/\lambda)=l_0 \tag{31}$$

となり, これを最単純モデルと呼ぶ.

e) タイプ $(1,S+1,1)$

バンク1のメモリに対して $S+1$ 段のストアバッファがありキャッシュをもつシステムである. ここに,

$$S=1,2,\cdots. \tag{32}$$

ここでは, 仮定(7)にかえ次の仮定を設定する.

仮定(9) $d+m_{aR}=m_{cR}$, $d+m_{aS}\leq m_{cS}$ (33)

図3, 4 は演算サイクルの終了時点(時点1)に着目したときの各事象である. 図3は演算サイクルが読

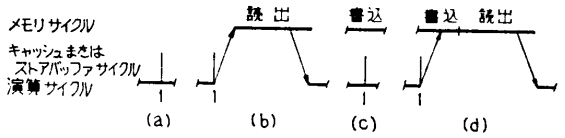

図3 読出メモリ要求で演算サイクル終了の場合
Fig. 3 Cases of issuing a read memory request at end of a processing cycle.

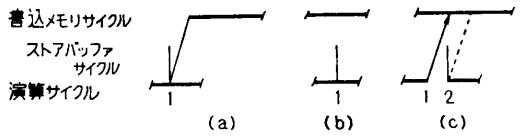

図4 書込メモリ要求で演算サイクル終了の場合
Fig. 4 Cases of issuing a store memory request at end of a processing cycle.

* 計算を実行すると

$$m_{cs}\sum_{i=0}^{n-1}\frac{(n-1)!}{(n-1-i)!}\left(\frac{\lambda\beta_s}{n\lambda_M}\right)^i\left[1-\frac{i+1}{\lambda_M m_{cs}}\left\{1-\left(1+\frac{\lambda\beta_s m_{cs}}{n}\right)^{n-1-i}\cdot\exp(-\lambda_M m_{cs})\right\}\right]$$

となる.

出メモリ要求で終了した場合で，(a)は書込メモリサイクル（以下では単にメモリサイクルという）中でなくキャッシュでヒットした場合，(b)はメモリサイクル中でなくキャッシュでミスヒットした場合である．(c)，(d)はメモリサイクル中であるが，(c)はキャッシュかストアバッファかでヒットした場合，(d)はキャッシュでもストアバッファでもミスヒットした場合である．図4は演算サイクルが書込メモリ要求で終了した場合で，(a)はそのときメモリサイクル中でなく，この事象でストアバッファのエントリ数は0から1に変化する．(b)はメモリサイクル中であるがストアバッファのエントリ数が $S+1$ より少ないため情報をストアバッファに入れて（エントリ数が1だけ増加する）次の演算サイクルを進める．(c)はストアバッファのエントリ数が $S+1$ になっているため実行中のメモリサイクルの終了時点2まで演算サイクルは停止し時点2から次の演算サイクルを開始するがストアバッファのエントリ数は依然 $S+1$ である．

ここで $\varphi(s,t)$ を求める．

まず，$1\leq s\leq S-1$ の場合を考えるが，この確率に寄与するのは図3(c)，(d)図4(b)，(c)の各場合である．これらの事象の時点1以降の最初の演算サイクルの終了時点が書込メモリサイクル中でかつストアバッファのエントリ数が $s+1$ である状態に着目して $\varphi(s,t)$ を求める．図3(c)の時点1でストアバッファのエントリ数が y_1+1 であるとその生起確率 P_1 は $-\beta_R(1-\alpha\alpha_S)d\varphi(y_1,\tau)/d\tau$ であり，時点1で始まった演算サイクルがそのとき実行中のメモリサイクル時間を t 以上残して終了するための遷移確率は

$$\int_t^{m_cs}d\tau\int_0^{\tau-t}\lambda\exp(-\lambda u)P_1du\quad(y_1=s)$$

であり（事象イ），このメモリサイクルから数えて連続実行された $i+2$ 番目のメモリサイクル時間を t 以上残して終了する遷移確率は

$$\int_0^{m_cs}d\tau\int_{\tau+im_cs}^{\tau+(i+1)m_cs-t}\lambda\exp(-\lambda u)P_1du$$
$$(y_1=s+i+1,\ i=0,1,\cdots,S-s-1)$$

である（事象ロ）．図3(d)の時点1でストアバッファのエントリ数が $s+i+2$ であるとその生起確率は $\beta_R\alpha\alpha_S\varphi(s+i+1,0)$ でありこのとき実行中のメモリサイクルの次は仮定(4)から読出メモリサイクルが行われひき続き連続実行された $i+1$ 個の書込メモリサイクルの最後のメモリサイクルを t 時間以上残して時点1以降で最初の演算サイクルが終了すると式(33)を考えてその遷移確率は $\exp(-\lambda im_cs)(1-\exp(-\lambda(m_cs-t)))$，$(i=0,1,\cdots,S-s-1)$ となる（事象ハ）．図4(b)の時点1の直前でストアバッファのエントリ数が y_2+1 であるとその生起確率 P_2 は $-\beta_Sd\varphi(y_2,\tau)/d\tau$ である．時点1で始まった演算サイクルが実行中のメモリサイクル時間を t 以上残して終了する遷移確率は

$$\int_t^{m_cs}d\tau\int_0^{\tau-t}\lambda\exp(-\lambda u)P_2du\quad(y_2=s-1)$$

であり（事象ニ），このメモリサイクルから数えて連続実行された $i+2$ 番目のメモリサイクル時間を t 以上残して終了する遷移確率は

$$\int_0^{m_cs}d\tau\int_{\tau+im_cs}^{\tau+(i+1)m_cs-t}\lambda\exp(-\lambda u)P_2du$$
$$(y_2=s+i,\ i=0,1,\cdots,S-s-1)$$

である（事象ホ）．図4(c)では時点1の直前でストアバッファのエントリ数が $S+1$ であるから時点1の生起確率は $\beta_S\varphi(S,0)$ でこのとき実行中のメモリサイクルから数えて連続実行された $S-s+2$ 番目のメモリサイクル時間を t 以上残して終了する遷移確率は $\exp(-\lambda(S-s)m_cs)(1-\exp(-\lambda(m_cs-t)))$ である（事象ヘ）．以上の事象イ～ヘの確率を加えると $\varphi(s,t)(1\leq s\leq S-1)$ が得られる．

$\varphi(0,t)$ ではストアバッファのエントリ数が1であるから事象ニは生起せず，事象イ～ハ，ホ，ヘで $s=0$ とした場合と，これに加えて図4(a)の場合がある．図4(a)の時点1の直前ではストアバッファのエントリ数は0で時点1の直後に1になるが，時点1で始まった演算サイクルが実行中のメモリサイクル時間を t 以上残して終了するための時点1の生起確率は $\varphi_0\beta_S$ で，遷移確率は $1-\exp(-\lambda(m_cs-t))$ である（事象ト）．

演算サイクルが終了したときストアバッファのエントリ数が $S+1$ であると $\varphi(S,t)$ が得られるから事象イ，ニ，ヘで $s=S$ とすればよい．

以上から $\varphi(s,t)(s=0,1,\cdots,S)$ を書き下すことができ，式(9)を考えて積分を実行し次式をうる．

$$\varphi(0,t)=\phi(0,t),\tag{34}$$

$$\varphi(s,t)=\phi(s,t)+\beta_S\left\{\exp(\lambda t)\int_t^{m_cs}\exp(-\lambda\tau)\cdot\frac{d}{d\tau}\varphi(s-1,\tau)d\tau+\varphi(s-1,\tau)\right\}$$
$$(s=1,2,\cdots,S),\tag{35}$$

ただし，$\phi(s,t)$ は次式で与えられる．

$$\phi(s,t)=a(s)(1-\exp(-\lambda(m_cs-t)))$$

$$+\beta_R(1-\alpha\alpha_S)\left\{\exp(\lambda\tau)\int_t^{m_{cs}}\exp(-\lambda\tau)\cdot\frac{d\varphi(s,\tau)}{d\tau}d\tau+\varphi(s,t)\right\}, \quad (36)$$

$$A(s)=-\beta_R(1-\alpha\alpha_S)\sum_{i=0}^{S-s-1}\exp(-i\lambda m_{cs})\cdot\int_0^{m_{cs}}\exp(-\lambda\tau)\frac{d}{d\tau}\varphi(s+i+1,\tau)d\tau-\beta_S\sum_{i=0}^{S-s-1}\exp(-i\lambda m_{cs})\int_0^{m_{cs}}\exp(-\lambda\tau)\cdot\frac{d\varphi(s+i,\tau)}{d\tau}d\tau+\beta_S\varphi(S,0)\exp(-(S-s)\lambda m_{cs})+\beta_R\alpha\alpha_S\sum_{i=0}^{S-s-1}\varphi(s+i+1,0)\exp(-i\lambda m_{cs}), \quad (37)$$

$$a(0)=\beta_S\varphi_0+A(0) \quad (38)$$

$$a(s)=A(s),\ (s=1,2,\cdots,S-1), \quad (39)$$

$$a(S)=\beta_S\varphi(S,0). \quad (40)$$

式(34), (35)を各々微分して得られた式と元の式とで積分項を消去すると1階 $S+1$ 元の微分方程式が得られこの一般解は式(4)を考慮して次式となる.

$$\varphi(s,t)=\sum_{r=0}^{s}(-\lambda\beta_S)^r\left\{\varphi(s-r,0)\frac{t^r}{r!}\exp(\lambda_M t)-\lambda a(s-r)\int_0^t\frac{t^r}{r!}\exp(\lambda_M t)dt\right\}\quad(s=0,1,\cdots,S). \quad (41)$$

これに式(9)の境界条件を入れると次式をうる.

$$\sum_{r=0}^{s}\frac{(-\beta_S\lambda m_{cs})^r}{r!}\varphi(s-r,0)\exp(\lambda_M m_{cs})=\lambda\sum_{r=0}^{s}(-\beta_S\lambda)^r a(s-r)\int_0^{m_{cs}}\frac{t^r}{r!}\exp(\lambda_M t)dt\quad(s=0,1,\cdots,S) \quad (42)$$

式(38), (39)に(41)を入れると式(11), (38)～(40), (42)で $a(s)$, $\varphi_0, \varphi(s,0)$ に関して $2S+3$ 個の連立方程式が得られこれらの未知数を確定できる. また, 式(14)に式(41)を入れて $\tilde{\omega}(s)$ が得られる.

キャッシュでも（ストアバッファにエントリがあるとき）ストアバッファでもミスヒットしたときはじめて実効読出アクセス時間がきき, 演算サイクル終了時メモリサイクル中でキャッシュとストアバッファでミスヒットした読出かストアバッファが満杯のときの書込かのメモリ要求でメモリ待を生ずるから演算サイクル当りの平均処理時間とメモリ使用率は次式となる.

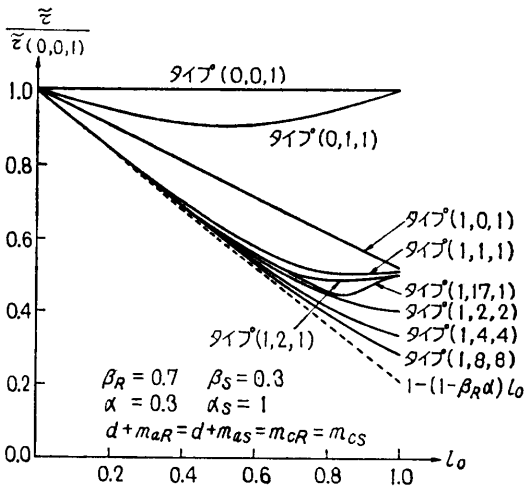

図5 各タイプの $\tilde{\tau}/\tilde{\tau}_{(0,0,1)}$ の例
Fig. 5 An example of $\tilde{\tau}/\tilde{\tau}_{(0,0,1)}$.

$$\tilde{\tau}_{(1,s+1,1)}=1/\lambda+\beta_R\alpha(d+m_{aR})\left(\varphi_0+\alpha_s\sum_{s=0}^{S}\varphi(s,0)\right)+\beta_R\alpha\alpha_S\sum_{s=0}^{S}\varphi(s,0)\tilde{\omega}(s)+\beta_S\varphi(S,0)\tilde{\omega}(S) \quad (43)$$

$$l_{(1,s+1,1)}=(\beta_R\alpha\alpha_S\zeta m_{cR}+\beta_S m_{cS})/\tilde{\tau}_{(1,s+1,1)} \quad (44)$$

2.4 各タイプの性能比較

各タイプの性能改善の傾向をとらえやすくするために, メモリサイクルとアクセスの各時間を式(29)に設定する. この条件によりタイプ(0, 0, 1)は式(30), (31)で与えられる最単純モデルになるが, **図5**はこの最単純モデルに対する各タイプの処理時間の比を示している. 横軸はメモリサイクル時間と演算サイクル時間の相対値で決まる式(17)の l_0 を使用している. l_0 は最単純モデルのメモリ使用率でもあるが(式(31)), メモリサイクル時間と演算サイクル時間のあらゆる変化を $0\leq l_0\leq 1$ の範囲で表わすパラメータの意味をもつ. 図5では $\beta_R=0.7$, $\beta_S=\alpha=0.3$ を例にとったが, これは事務計算の環境でワーストケースに近くメモリ待がより顕著の方向にある. 点線はメモリ待がないとした場合（式(20)で $I(m_{cs},\alpha,n)=0$）でこれが前記条件での性能改善の限界である. 図5から次のことがわかる.

- ストアバッファは l_0 が1または0に近くない（演算サイクル時間とメモリサイクル時間とがバランスしている）領域で効果が大きい.

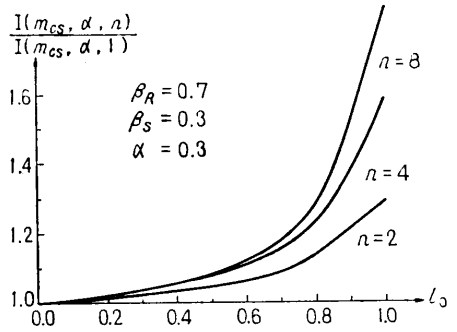

図 6 $I(m_{cs},\alpha,n)/I(m_{cs},\alpha,1)$ の例
Fig. 6 An example of $I(m_{cs},\alpha,n)/I(m_{cs},\alpha,1)$.

● キャッシュは l_0 が1に近づく（メモリサイクル時間が演算サイクル時間に比べて大きくなる）につれ効果が大きくなりその度合はストアバッファ，メモリインタリーブの効果より大きい．

● メモリインタリーブは l_0 が小さい（演算サイクル時間がメモリサイクル時間に比べ大きい）とき効果がなく l_0 が1に近づくにつれ顕著な効果を示し，ストアバッファの性能改善の弱点を補っている．

● ストアバッファの段数効果は l_0 が大きいところでのメモリインタリーブの効果に劣るが，いずれもそれらの数の増加とともに性能改善への寄与率は大幅に低下する．

また図5にはないが，タイプ (0, 1, 1) を除き一般に $\beta_R\alpha+\beta_S$ を小さくすると図5上での各タイプの曲線の傾斜を大きくでき，更にメモリインタリーブしないタイプでは性能改善のピークを $l_0=1$ の方向に移動させる効果をもつ．α の効果から β_R を大きくすると $\beta_R\alpha+\beta_S$ が小さくなる．しかし，逆にタイプ (0, 1, 1) では β_R が大きくなると性能が低下してゆく．

図6は式(21)の $I(m_{cs}, \alpha, n)$ を $n=1$ の場合との比で示している．同図から l_0 が小さくなるにつれ平均メモリ待時間を $\beta_S I(m_{cs}, \alpha, 1)/n$（メモリインタリーブしない場合の $1/n$）で近似できること，l_0 が1に近くなるにつれバンク当りの効果が小さくなることを示している．

3. 多重プロセッサシステムの解析

3.1 近似モデルによる性能解析

p 個のプロセッサがバンク数 n のインタリーブされた1つのメモリを共用するシステムを考える．複数メモリユニットの場合は系の総バンク数を n とする．

プロセッサ h $(h=1, 2, \cdots, p)$ は，単一プロセッサシステムで演算サイクル当りの平均処理時間 $\tilde{\tau}^{(h)}$，平均命令実行時間 $\tilde{\imath}^{(h)}$，メモリ使用率 $l^{(h)}$，命令当りの平均メモリ要求回数 $\tilde{\gamma}^{(h)}$，平均メモリサイクル時間 $m^{(h)}$ をもち，多重プロセッサシステムでのメモリ使用率を $L^{(h)}$，平均命令実行時間を $\tilde{T}^{(h)}$，単一に対する多重プロセッサシステムでの性能比を $\rho^{(h)}$，単一プロセッサシステム群に対する多重プロセッサシステムのスループット比（多重プロセッサ性能係数）を ψ とする．$\rho^{(h)}$ は skinner[5]の stretching factor の逆数に相当する．

$$\rho^{(h)}=L^{(h)}/l^{(h)},\ (h=1,2,\cdots,p), \tag{45}$$

$$\psi=\rho^{(1)}+\rho^{(2)}+\cdots+\rho^{(p)}, \tag{46}$$

$$\tilde{T}^{(h)}=\tilde{\imath}^{(h)}/\rho^{(h)}=\tilde{\gamma}^{(h)}\tilde{\tau}^{(h)}/\rho^{(h)},\quad (h=1,2,\cdots,p). \tag{47}$$

多重プロセッサシステムでは，図2(a)，(b)の時点 i' で発行したメモリ要求がほかのプロセッサのメモリサイクルで待たされることがあるため時点 $i+1$ の待確率が減少する方向にあるが，図6から時点 i' でメモリ要求が発行される確率は小さいとみられ次を仮定する．

仮定(10) 任意のプロセッサが自分で発行したメモリサイクル上での待時間は多重プロセッサシステムでも単一プロセッサシステムとかわらない．

2 では読出・書込のメモリサイクルと演算サイクルの時間関係で各種単一プロセッサシステムの処理時間の解析を行ったが，多重プロセッサシステムにおいては1つのプロセッサがほかのプロセッサの性能に与える影響はメモリ要求の頻度とそのサイクル時間である．

仮定(11) 任意のプロセッサからの任意のメモリ要求でのバンク指定はランダムである．

この仮定から，プロセッサ h のバンク k のメモリ使用率を $l^{(h)}{}_k$ とすると次式が成立する．

$$l^{(h)}{}_k=l^{(h)}\ (k=1,2,\cdots,n) \tag{48}$$

ゆえに，n バンク上でのメモリ競合は1バンク上でのそれと等価な確からしさをもつから，等価的に平均演算サイクル時間 $1/\lambda^{(h)}$，平均メモリサイクル時間 $m^{(h)}$，メモリ使用率 $l^{(h)}$ で1つのバンクのみにアクセスする **2.** での最単純モデルとみなせる．今，$m^{(h)}$ とメモリからみた演算サイクル時間 U の確率 $F^{(h)}(U)$ を次で仮定する．

仮定(12) $F^{(h)}(U)=\lambda^{(h)}\exp(-\lambda^{(h)}U)$ (49)

仮定(13) $m^{(h)}=m$

したがって，n バンクのメモリを共用した p プロセ

ッサシステムは単一バンクのメモリを共用する p 個の最単純モデルのプロセッサからなるシステムで近似できる．この $p=2$ の場合については筆者が解析している[2),3)]．また，各プロセッサの演算サイクルのパラメータがすべて同じ p プロセッサシステムではTAKÁCS[4)]の待時間解析の結果がそのまま使用できるが複雑な式となる．各プロセッサの特性が異なる場合の p プロセッサシステムも解析できるが一段と複雑になる．そこで，$p=2$ の筆者の結果を使用した p プロセッサシステムの近似解法を考える．

プロセッサ $1\sim p_1$ からなる p_1 プロセッサシステム(グループ x_1)とプロセッサ $p_1+1\sim p_1+p_2$ からなる p_2 プロセッサシステム(グループ x_2)が独立にあり，その各メモリ使用率を $L^{(x_1)}$ と $L^{(x_2)}$ とすると

$$L^{(x_1)}=L^{(1)}+L^{(2)}+\cdots+L^{(p_1)}\leq 1 \quad (50)$$

$$L^{(x_2)}=L^{(p_1+1)}+L^{(p_1+2)}+\cdots+L^{(p_1+p_2)}\leq 1 \quad (51)$$

であるから，前記と同様な近似で，グループ x_i をメモリからみた平均演算サイクル時間 $1/\Lambda^{(x_i)}$，メモリ使用率 $L^{(x_i)}$，メモリサイクル時間 m をもち単一バンクのメモリにアクセスする単一プロセッサシステムとみなし，演算サイクル時間 U の確率 $F^{(x_i)}(U)$ を次で仮定する．

仮定(14) $F^{(x_i)}(U)=\Lambda^{(x_i)}\exp(-\Lambda^{(x_i)}U)$ (52)

グループ x_1 と x_2 を1つの p_1+p_2 プロセッサシステムに組んだとき，グループ x_i のメモリサイクル当りの平均メモリ待時間 $\tilde{\omega}^{(x_i)}$ とメモリ使用率 $L^{(x_i)'}$ はメモリ使用率 $L^{(x_i)}$ のみで定まる次式で与えられる[2),3)]．

$$L^{(x_i)}/L^{(x_i)'}=1+L^{(x_i)}\cdot\tilde{\omega}^{(x_i)}/m \quad (53)$$

$$L^{(x_i)}=m/(m+1/\Lambda^{(x_i)}) \quad (54)$$

$$\left.\begin{aligned}\tilde{\omega}^{(x_1)}/m&=(L^{(x_2)'}/L^{(x_1)'})K^{(x_1)}\\ \tilde{\omega}^{(x_2)}/m&=(L^{(x_1)'}/L^{(x_2)'})K^{(x_2)}\end{aligned}\right\} \quad (55)$$

$$K^{(x_i)}=1-(1/\Lambda^{(x_i)}m)(1-\exp(-\Lambda^{(x_i)}m)) \quad (56)$$

仮定(14)は，グループ x_i 内のプロセッサ h_i のメモリサイクル当りの待時間 $\tilde{\omega}^{(h_i)}$ にグループ内のほかのプロセッサのメモリ待時間が加算されることを意味するから

$$\tilde{\omega}^{(h_i)}=(L^{(x_i)'}/L^{(h_i)'})\cdot\tilde{\omega}^{(x_i)} \quad (57)$$

で与えられる．ここに $L^{(h_i)'}$ は p_1+p_2 プロセッサシステムでのプロセッサ h_i のメモリ使用率で，式(53)(54)の x_i を h_i で置きかえて与えられ，これに，式(53)，(57)を代入すると次式が得られる．

$$L^{(h_i)'}/L^{(h_i)}=L^{(x_i)'}/L^{(x_i)} \quad (58)$$

したがって，システムのスループット ψ' が次式で得られる．

$$\psi'=\sum_{h_1=1}^{p_1}L^{(h_1)'}/l^{(h_1)}+\sum_{h_2=p_1+1}^{p_1+p_2}L^{(h_2)'}/l^{(h_2)} \quad (59)$$

以上から，$p_1=p_2=1$ から始めて，順次多重プロセッサの解が得られる．

この近似手法で得られた ψ' より実際の値が小さくなることはないが，若干細かくみると，仮定(14)は $L^{(x_i)}$ が1に近づくにつれ確からしいが，0に近づくにつれ $\tilde{\omega}^{(h_i)}$ は $\tilde{\omega}^{(x_i)}$ に近づくから式(53)の $L^{(x_i)}$，$L^{(x_i)'}$ を $L^{(h_i)}$，$L^{(h_i)'}$ でおきかえて次式が得られる．

$$L^{(h_i)'}/L^{(h_i)}<1/(1+L^{(h_i)}\tilde{\omega}^{(x_i)}/m) \quad (60)$$

式(60)は $L^{(x_i)}$ が小さいとき確からしいが，このとき $L^{(h_i)}$，$\tilde{\omega}^{(x_i)}$ ともに小さくなり式(58)と(60)による $L^{(h_i)'}$ の差は微小となる．**3.2** でこれを評価する．また，個々の $L^{(h_i)'}/l^{(h_i)}$ より ψ' に着目した方が近似度が高い．

3.2 多重プロセッサ性能係数 ψ の評価

図7は各プロセッサの単一プロセッサシステムでのメモリ使用率が l である p プロセッサシステムの ψ の例である．実線は式(58)，点線は式(60)の場合である．

pl が小さいところでは実線と点線の差は微小である．pl が大きくなると差が開いてくるが，当然 $\psi\leq 1/l$ であるから実線は実態に近いとみてよい．

性能の観点からすると，$p=2$ は実用性があるといえるが，$p=4$ になると $l=0.2$ 以下が望ましい．

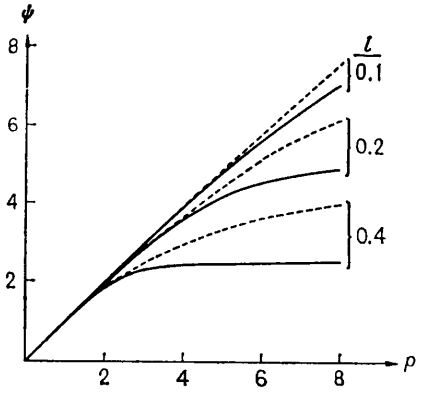

図7 ψ の例
Fig. 7 An example of ψ.

4. 各パラメータの性格と問題処理過程の近似

4.1 各パラメータの性格

プロセッサ性能を決める主たる要因としてハードウ

ェアテクノロジ(A)，ハードウェア論理構造(B)，ソフトウェア構造(C)とシステムアーキテクチャ(D)があるが，これらと **2.** で定義した各パラメータとの関連を若干考察する．

$\beta_R, \beta_S, \tilde{\gamma}$；命令の定義(D)とプログラムでの命令出現頻度(C)に依存し，更にメモリ読出／書込幅等(B)にも関係する．

λ：プロセッサの論理構造(B)とクロック周期時間(主としてA)に大きく依存し，命令の定義(D)とそれのプログラムでの出現頻度(C)にも関係する．先行制御の乱れ(B)もこの中に含まれる．

$m_{cR}, m_{cS}, m_{aR}, m_{aS}, d$；AとBに依存する．全書込と部分書込でメモリサイクル時間が異なる場合等は平均値で近似してよい．

α；メモリとキャッシュとのマッピング方式，キャッシュの置換アルゴリズムと容量等のキャッシュ構造(B)に依存しプログラムのアドレスパタン(C)にも関係する．

α_S；プログラムのアドレスパタンに依存し(C)ストアバッファ量(B)に関係する．

ζ, n, s；ハードウェア論理構造(B)できまる．

4.2 問題の処理過程の近似とパラメータ値の設定

問題の処理過程はプロセッサ上を走行する実際のプログラムの動的な流れであるから，性能を評価したいプログラムを机上でシミュレーションするか，特定のプロセッサ上で走行させながら，ソフトウェアシミュレータかハードウェアモニタで $\tilde{\gamma}, \beta_R, \beta_S, \alpha, \alpha_S$ と命令の種類別出現頻度（λ を決める）を知ることができる．これらのパラメータはメモリの読出／書込幅，キャッシュ／ストアバッファの容量等で変化するから設計目標の論理構造に合わせてパラメータ値を近似修正し，ハードウェアテクノロジか論理構造で決まるその他のパラメータ値を設定し，**2, 3** の理論式を使用して設計目標とするプロセッサ系の性能を予測することができる．結果に満足できないときは，ハードウェア論理構造，ハードウェアテクノロジ，ソフトウェア構造，システムアーキテクチャのいずれかを修正する(**図8**)．問題の処理過程がすでにジョブミックスの形で与えられているときはシミュレーションによらずこのジョブミックスから $\tilde{\gamma}, \beta_R, \beta_S, \lambda$ を知ることができる．α, α_S は類似の環境下でのシミュレーション結果または経験的に得られているデータ等をベースに近似する．ジョブミックスは古くから用いられているSGM, BGM があるが最近では個別オペレーティング

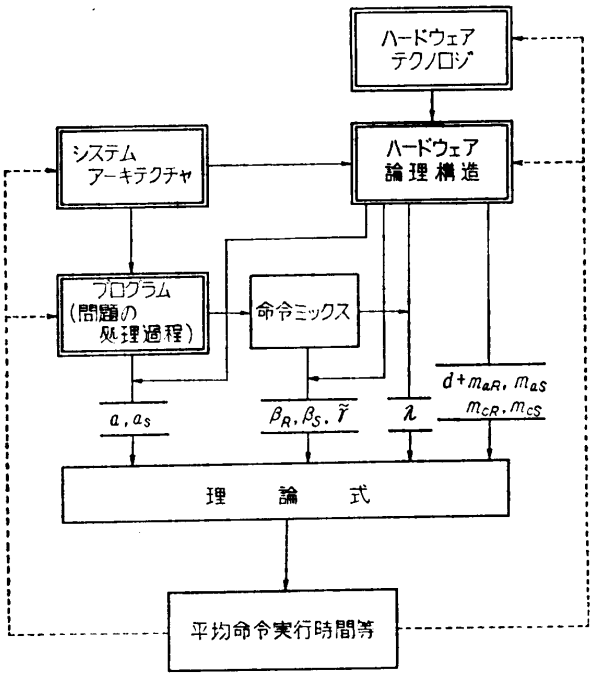

図 8 本理論適用のフローチャート
Fig. 8 An application flow chart for this theory.

システム環境下で問題別に各種ミックスが考えられハードウェアのシステム設計の参考に供しられている．

5. 本理論とシミュレーション結果との比較

シミュレーションモデルは，同じ特性をもつ中央処理装置 (CPU) とデータチャネル装置 (DCH) の各最大2台がメモリインタフェースユニット (SIU) に接続され，4ウェイインタリーブされたメモリユニット (MMU) に最大2台の SIU が接続されるシステムである．CPU が 1, 2, 4 台の各ケースについてシミュレーションを行ったが，CPU と同じ数だけ DCH がつき，2SIU は4CPU のときのみである．

情報の流れに着目した CPU, SIU, MMU のブロック図は**図9**である．太線の四角はレジスタで単相クロックに同期している．CPU はパイプライン化された先行制御がなされている．オペランドアドレス生成ブロックは内部に1段レジスタをもち論理動作に2クロックを要する．演算回路は演算を実行するブロックで当然内部に多数のレジスタをもっている．TLB (Translation Lookaside Buffer) は論理アドレスをメモリの物理アドレスに変換するブロックである．MMU のインタリーブはクロック周期時間 (μ) の位相だけずれて行われる．MMU からキャッシュへの情報転送は4回行われる ($\zeta=4$)．メモリは 346μ ごとに4バン

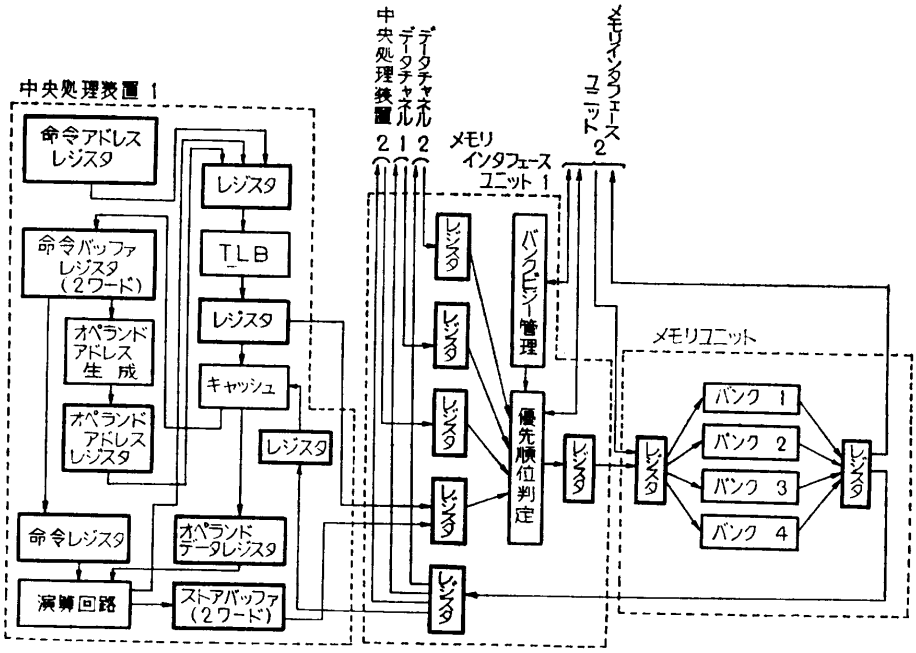

図 9 シミュレーションモデル
Fig. 9 The simulation model.

表 1 パラメータ値
Table 1 Values of parameters.

ケース	CPU数	α	$\tilde{\tau}$	β_R	β_{sf}	β_{sp}
1	1	0.052	2.442	0.889	0.102	0.009
2	2	0.052	2.453	0.886	0.105	0.009
3	4	0.051	2.455	0.890	0.100	0.010

$m_{aR}=m_{cR}=m_{csf}=d=5\mu$, $m_{csp}=11\mu$, $\tilde{\tau}\beta_R\alpha=0.11$, $\zeta=n=4$

表 2 シミュレーション結果との比較
Table 2 Comparison of results with this theory to simulation.

ケース	計算値				シミュレーション値	
	$1/\lambda$	l	$\tilde{t}$	$\tilde{T}$	$\tilde{t}$	$\tilde{T}$
1	1.853^{μ}	0.235	5.77^{μ}	$-^{\mu}$	5.59^{μ}	$-^{\mu}$
2	1.845	0.165	5.74	5.96	—	5.89
3	1.843	0.163	5.72	6.71	—	6.64

ク同時のリフレッシュを行う．DCH 当り $(160/9)\mu$ に1回の割合でメモリ要求を出す．

CPU を走行するプログラムは主としてバンキングジョブに相当するジョブミックスを与え，$\tilde{\tau}\beta_R\alpha=0.11$ とした．

シミュレーションは前記モデルを GPSS で記述し CPU 数 1, 2, 4 の3ケースにつき 10^4 命令を実行し表1の $\alpha, \tilde{\tau}, \beta_R, \beta_{sf}, \beta_{sp}$ と表2のシミュレーション値の $\tilde{t}, \tilde{T}$ を得た．ただし，β_{sf}, β_{sp} と m_{csf}, m_{csp} はそれぞれ全書込み，部分書込の確率とサイクル時間で

次式となる．

$$\beta_s=\beta_{sf}+\beta_{sp}. \tag{61}$$

また，$\tilde{t}$, $\tilde{T}$ はそれぞれ単一と多重 CPU システムにおける CPU の平均命令実行時間である．

本理論による計算値はタイプ (1, 4, 4) とそれの多重プロセッサシステムである．$\alpha=0$ とした単一 CPU システムのシミュレーションで得られた平均命令実行時間（メモリ待時間は無視できる小さな値であった）と表1の $\tilde{\tau}$ を使用して式(20)から λ を求め，これと表1の各パラメータ値を使用して式(15), (20), (22)から $\tilde{t}$ と単一 CPU システムでのメモリ使用率 l を求め，式(47), (53)～(56), (58)から $\tilde{T}$ を求めた．ただし，m_{cs} は

$$m_{cs}=(\beta_{sf}m_{csf}+\beta_{sp}m_{csp})/\beta_s \tag{62}$$

とした．また，DCH のメモリ要求とメモリリフレッシュはまとめて相当するメモリ使用率をもつ1つのプロセッサとして扱った．したがって，ケース 1, 2, 3 は，実は，2, 3, 5 プロセッサシステムの例となる．

表2にみられるように，計算値はシミュレーション値に対し＋1～3%の差できわめてよく一致している．

6. むすび

キャッシュ，ストアバッファとメモリインタリーブの有無で構成される各種単一プロセッサシステムのう

ち小型から大型コンピュータで実際に近いと想定される5つのタイプについて，ハードウェアテクノロジ，ハードウェア論理構造，ソフトウェア構造とシステムアーキテクチャの特性を表わす若干のパラメータを使用してプロセッサ性能を解析し，更に単一プロセッサシステムでのメモリ使用率がわかっているときプロセッサ数が任意の多重プロセッサシステムの性能を近似的に解析することができ，これらがシミュレーション結果とよく一致することを確かめた．

λ の中には CPU のパイプライン制御の乱れも含まれるが，これの評価理論は本理論の形を崩さず定式化できると考えている．

謝辞 本研究のご指導をいただいた静岡大学工学部松本欣二教授他諸先生，機会と激励をいただいた日本電気株式会社宮城嘉男支配人と金井久雄コンピュータ技術本部長，シミュレーションデータを提供された同社仙波清主任に深謝する．

参考文献

1) 齋藤将人：マイクロコマンド分布を利用した中央処理装置の処理時間解析，信学論(C)，54-C, 4, pp. 285-293 (昭 46-04).
2) 齋藤将人：デュアルプロセッサシステムのメモリ待ち時間の一般的解法とその応用例について，信学論(D)，55-D, 2, pp. 83-90 (昭 47-02).
3) 齋藤将人：メモリ競合が2プロセッサシステムの性能に与える影響について，信学論(D)，58-D, 11, pp. 657-664 (昭 50-11).
4) TAKÁCS, L.: On a Stochastic Process Concerning some Waiting time Problems, Theory of Probability and its Applications, Vol. II, No. 1, pp. 90-103 (1957).
5) Skinner, C. E. and Asher, J. R.: Effects of Storage Contention on System Performance, IBM Syst. J., 8, 4, pp. 319-333 (1969).

(昭和54年8月28日受付)
(昭和55年5月15日採録)