

次の問1から問7までの7問については、この中から5問を選択し、選択した問題については、答案用紙の選択欄の(選)をマークして解答してください。

なお、6問以上選択した場合には、はじめの5問について採点します。

問1 キャッシュメモリに関する次の記述を読んで、設問1, 2に答えよ。

キャッシュメモリとは、主記憶とCPUの間に置く高速アクセスが可能なメモリである。キャッシュメモリとCPU及び主記憶との関係を図1に示す。データをキャッシュメモリに保持しておくことによって、CPUは速度の遅い主記憶に直接アクセスしなくて済むので、処理の高速化が図れる。

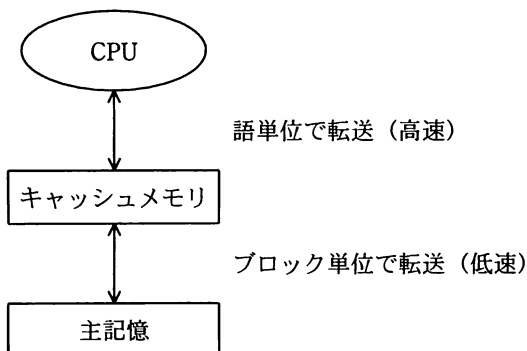


図1 キャッシュメモリとCPU及び主記憶との関係

ここでは、ハードウェアのアーキテクチャを次のように仮定する。

- (1) 主記憶はブロック（1ブロックは100語から成る）に分割されている。各ブロックには、その先頭番地が小さいものから順に1, 2, 3, …とブロック番号が振られている。主記憶とキャッシュメモリ間はブロック単位でデータが転送される。
- (2) キャッシュメモリには、命令を保持しておく命令キャッシュと、データを保持しておくデータキャッシュの2種類がある。ここでは、データキャッシュ（以下、キャッシュという）だけを考える。
- (3) キャッシュの構成は、図2のとおりとする。
 - ① キャッシュは、ディレクトリ部とデータ部から成る。
 - ② データ部はバッファ1～3の三つのバッファから成り、各バッファは1ブロック分の主記憶の内容を保持できる。

③ ディレクトリ部は、データ部のバッファ1～3に対応したディレクトリ1～3から成る。それぞれのディレクトリは次の内容を保持する三つのフィールドから成る。

なお、データ部のバッファが未使用の場合は、対応するディレクトリの三つのフィールドすべてに0が入っている。

(イ) ブロック番号：対応するデータ部のバッファが保持する主記憶のブロック番号

(ロ) 順位：キャッシュ内に最も古くから存在するブロックから順に1, 2, 3と番号が振られる。

(ハ) フラグ：対応するデータ部のバッファにブロックを読み込んだとき、0に初期化される。対応するデータ部のバッファに保持されている内容がCPUの処理によって変更されると、1に変わる。

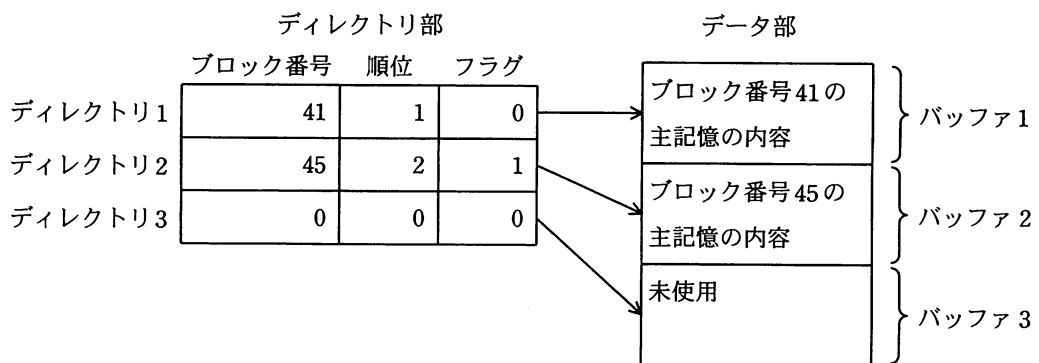


図2 キャッシュの構成

擬似言語で表現された次の繰返し処理を実行する場合について考える。

[繰返し処理]

```

■ i: 99, i ≥ 0, -1
  |
  |   ・ A[i] ← A[i] + B[i] + C[i]
  |   ・ D[i] ← A[i]
  |
■

```

(1) 配列 A, B, C 及び D は 100 個の要素から成り、1 要素は 1 語である。添字は 0 から始まるものとする。

(2) データ領域の主記憶への割付けは、次のとおりとする。

① Aの配列領域：4000～4099番地（ブロック番号41）

A[0] は4000番地，A[1] は4001番地という順に割り付けられる。

② Bの配列領域：4100～4199番地（ブロック番号42）

③ Cの配列領域：4200～4299番地（ブロック番号43）

④ Dの配列領域：4300～4399番地（ブロック番号44）

⑤ 定数-1と99の格納領域：4400，4401番地（ブロック番号45）

(3) 変数*i*はレジスタを使用し，主記憶への割付けは行わない。

(4) プログラム領域の内容は表1のとおりとする。参照ブロックの数值は，その命令を実行するときCPUが参照するデータのブロック番号を示す。

表1 プログラム領域の内容

番地	命令	処理の内容	参照ブロック
1000	LOAD R1,4400	-1をR1に設定	45
1001	LOAD R2,4401	<i>i</i> の初期値(99)をR2に設定	45
1002	LOAD R3,4000,R2	A[<i>i</i>]の内容をR3に設定	41
1003	ADD R3,4100,R2	B[<i>i</i>]の内容をR3に加算	42
1004	ADD R3,4200,R2	C[<i>i</i>]の内容をR3に加算	43
1005	STORE R3,4000,R2	R3の内容をA[<i>i</i>]に格納	41
1006	STORE R3,4300,R2	R3の内容をD[<i>i</i>]に格納	44
1007	ADDR R2,R1	$i \leftarrow i - 1$	-
1008	JNM 1002	$i \geq 0$ ならば1002番地にジャンプ	-

命令の形式は次のとおりとする。

LOAD / ADD / STORE Rn , adr [, Rx] :

定数 *adr* はアドレスを示す。Rxは指標レジスタである（省略可能）。Rxを指定した場合の実効アドレスは，定数 *adr* と Rx の内容を加算した値が示す番地とする。

LOAD は，実効アドレスが示す番地に格納されている内容を，レジスタ Rn に設定する。

ADD は，レジスタ Rn の内容に実効アドレスが示す番地に格納されている内容を加えて，レジスタ Rn に設定する。

STORE は，レジスタ Rn の内容を実効アドレスが示す番地に格納する。

ADDR Rn , Rm : レジスタ Rn の内容にレジスタ Rm の内容を加えて，レジスタ Rn に設定する。

JNM adr : 直前の演算結果が0以上ならば *adr* 番地へジャンプする。

設問1 次の(1)～(6)に従って、1000番地のLOAD命令を実行した直後、及び1006番地のSTORE命令を最初に実行した直後のディレクトリ部の内容を、図3と図4に示す。 に入れる正しい答えを、解答群の中から選べ。

- (1) 参照ブロックがキャッシュ内にある場合は、キャッシュ内のデータが使用される。
- (2) 参照ブロックがキャッシュ内がない場合（以下、ミスという）は、参照ブロックが主記憶からキャッシュに読み込まれ、対応するディレクトリのフラグの内容は0に初期化される。
- (3) CPUが参照ブロックに対してSTORE命令を実行した場合は、対応するディレクトリのフラグの内容は1に変わる。
- (4) ミスが起こったときにキャッシュ内に未使用のバッファがある場合は、未使用のバッファの中で最も番号が小さいバッファに参照ブロックを読み込み、順位を更新する。
- (5) ミスが起こったときにキャッシュ内に未使用のバッファがない場合は、次のキャッシュ更新ロジックを用いる。
キャッシュ内に最も古くから存在するブロックが格納されているバッファに、参照ブロックを読み込み、順位を更新する。ただし、対応するディレクトリのフラグの内容が1だったときは、そのブロックを主記憶に書き戻してから、参照ブロックを読み込み、順位を更新する。
- (6) プログラム実行開始時は、キャッシュ内にデータが入っていないものとする。

	ブロック番号	順位	フラグ
ディレクトリ1	a	1	0
ディレクトリ2	0	0	0
ディレクトリ3	0	0	0

図3 1000番地のLOAD命令を実行した直後のディレクトリ部

	ブロック番号	順位	フラグ
ディレクトリ1	b	2	0
ディレクトリ2	44	3	1
ディレクトリ3	c	1	0

図4 1006番地のSTORE命令を最初に実行した直後のディレクトリ部

解答群

ア 0

イ 41

ウ 42

エ 43

オ 44

カ 45

設問2 設問1の(5)のキャッシュ更新ロジックを、次に示す新しいキャッシュ更新ロジックに変えた場合の、ディレクトリ部のブロック番号とフラグの内容の変化を、表2に示す。□に入れる正しい答えを、解答群の中から選べ。

[新しいキャッシュ更新ロジック]

参照されていない時間が最も長いブロックが格納されているバッファに、参照ブロックを読み込む。

なお、この更新ロジックでは、順位の付け方も変更されていて、キャッシュ内で参照されていない時間が最も長いブロックから順に1, 2, 3と番号が振られる。

表2 ディレクトリ部の内容の変化

実行命令の番地	ディレクトリ1		ディレクトリ2		ディレクトリ3	
	ブロック番号	フラグ	ブロック番号	フラグ	ブロック番号	フラグ
1000	45	0	0	0	0	0
1001	45	0	0	0	0	0
1002	45	0	41	0	0	0
1003	45	0	41	0	42	0
1004	43	0	41	0	42	0
1005	43	0	41	1	42	0
1006			d			
1007						
1008						
1002					e	
1003	f					
1004						
1005						
1006						
1007						
1008						

注 網掛けの部分は、表示していない。

解答群

	ブロック番号	フラグ
ア	41	0
イ	41	1
ウ	42	0
エ	42	1
オ	43	0
カ	43	1
キ	44	0
ク	44	1
ケ	45	0
コ	45	1