

【解答】

【設問 1】 aーカ, bーエ, cーウ

【設問 2】 dーイ, eーク, fーウ

【解説】

キャッシュメモリに関するハードウェアの問題である。テーマとしては、ページングアルゴリズムで重要な FIFO (First In First Out) 及び LRU (Least Recently Used) の内容を命令の実行と絡めて問うものである。

対策として、その考え方を学習していた受験者が多いと推測するが、問題文が長く、解答内容の理解に時間を要すること、命令の理解と実行内容を確認しながらのトレース (追跡) が必要であることから、制限時間内に解くという点では難易度は高いといえる。

設問 1 は検証、確認が比較的容易にできるはずであるので、ここで基礎点を獲得しておきたい。設問 2 は命令の理解と実行内容を確認しながらのトレースにやや時間がかかるが、落沉着いて解き、ミスをしないよう気を付ける必要がある。

キャッシュを構成するデイレクトリ部、データ部の内容は問題文の図 2 に示されているが、デイレクトリ部のブロック番号、順位、フラグの意味を正確に把握することが解決のポイントである。

問題文の表 1 で示されているプログラム領域 (命令) とデータ領域の内容、及びプログラムが行う処理内容の概要を示し、各設問について解説する。なお、説明の便宜上命令番号①～⑨を付与しており、プログラムの実行に伴う段階的な値を P1～P3 としている。

1. 命令とプログラム領域の内容

表 A プログラム領域の内容

命令番号	番地	命令	処理の内容	参照ブロック
①	1000	LOAD R1, 4400	－1 を R1 に設定	45
②	1001	LOAD R2, 4401	i の初期値(99)を R2 に設定	45
③	1002	LOAD R3, 4000, R2	A[i] の内容を R3 に設定	41
④	1003	ADD R3, 4100, R2	B[i] の内容を R3 に加算	42
⑤	1004	ADD R3, 4200, R2	C[i] の内容を R3 に加算	43
⑥	1005	STORE R3, 4000, R2	R3 の内容を A[i] に格納	41
⑦	1006	STORE R3, 4300, R2	R3 の内容を D[i] に格納	44
⑧	1007	ADDR R2, R1	i ← i － 1	－
⑨	1008	JNM 1002	i ≧ 0 ならば 1002 番地にジャンプ	－

2. データ領域の主記憶への割付け内容 (各配列領域の添字は 0～99 が対応する)

配列領域の添字 i →	0	1	...	98	99
番地 →	4000	4001	...	4098	4099
A の配列領域 (ブロック番号 41)	A[0]	A[1]	...	A[98]	A[99]
番地 →	4100	4101	...	4198	4199
B の配列領域 (ブロック番号 42)	B[0]	B[1]	...	B[98]	B[99]
番地 →	4200	4201	...	4298	4299
C の配列領域 (ブロック番号 43)	C[0]	C[1]	...	C[98]	C[99]
番地 →	4300	4301	...	4398	4399
D の配列領域 (ブロック番号 44)	D[0]	D[1]	...	D[98]	D[99]
番地 →	4400	4401
定数－1 と 99 の格納領域 (ブロック番号 45)	－1	99

3. プログラムが行う処理内容

i : 99, i ≧ 0, －1	...	処理 1
・ A[i] ← A[i] + B[i] + C[i]	...	処理 1
・ D[i] ← A[i]	...	処理 2

疑似言語で表現されているが、処理は添字の初期値を 99 として、－1 しながら配列領域 A～C の要素を加算していく内容である。加算した値は、配列領域 A と D に格納されていく。

i の初期値 99 の場合における処理 1, 2 の段階別内容

- (1) P1 ← A[i] + B[i] = A[99] + B[99]
- (2) P2 ← P1 + C[i] = A[i] + B[i] + C[i]
- = A[99] + B[99] + C[99]

【設問 1】

- 空欄 a～c：命令の実行に伴いデイレクトリ部の内容がどうなるかは、設問(1)～(6)に記述されているので、よく把握しながら解けばよい。なお、キャッシュ更新ロジックは FIFO (最も古くから存在するブロックと置き換える) であり、置き換えが発生した場合は、順位 1 のデイレクトリ部と置き換える。順位の繰下げは－1 すればよい。

表 A に示した内容に基づき、トレースを行うと次のようになる。

命令番号①：1000 番地の命令「LOAD R1, 4400」を実行する。参照ブロック番号は 45 で、処理の内容は、「－1 を R1 に設定する」ものである。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 (設問 1(6)) こと、及び設問 1(1), (2) からデイレクトリ部の内容は図 3-1 のようになる。

R1	R2	R3	ブロック番号	順位	フラグ
－1			デイレクトリ部 1	45	1
			デイレクトリ部 2	0	0
			デイレクトリ部 3	0	0

順位は最も古くから存在することになり 1

図 3-1 1000 番地の LOAD 命令を実行した直後の各レジスタとデイレクトリ部

したがって、空欄 a は (カ) が正解である。

次に、1006 番地の STORE 命令を最初に行った直後のデイレクトリ部の内容を吟味してみる。以下、順にその様子を示す。

命令番号②：1001 番地の命令「LOAD R2, 4401」を実行する。参照ブロック

番号は 45 で、処理の内容は、「i の初期値(99)を R2 に設定する」ものである。これは命令番号①とほぼ等機能の内容で、参照ブロック番号 45 は既にキャッシュ内に存在するので、デイレクトリ部の内容に変化がない。なお、フラグは、対応するデータ部のバッファに保存されている内容が CPU の処理によって変更された場合 (STORE 命令実行時) に 1 と変化するが、ここでは変更は発生しないので 0 のままであり、R2 に 99 が設定される。

R1	R2	R3	ブロック番号	順位	フラグ
－1	99		デイレクトリ部 1	45	1
			デイレクトリ部 2	0	0
			デイレクトリ部 3	0	0

図 3-2 1001 番地の LOAD 命令を実行した直後の各レジスタとデイレクトリ部

命令番号③：1002 番地の命令「LOAD R3, 4000, R2」を実行する。この命令は R2 を指標レジスタとして利用している。実効アドレスは 4000 と R2 の内容を加算した値が示す番地となるので、4000 + 99(R2 の内容) = 4099 番地である。処理の内容は、「A[i] = 99」の内容を R3 に設定すること、参照ブロック番号は 41 になる。新たなブロックが参照されることになるため、デイレクトリ部 2 に順位を 2 として追加される。これらをまとめると、レジスタ及びデイレクトリ部は次のようになる。

R1	R2	R3	ブロック番号	順位	フラグ
－1	99	A[99]	デイレクトリ部 1	45	1
			デイレクトリ部 2	41	2
			デイレクトリ部 3	0	0

図 3-3 1002 番地の LOAD 命令を実行した直後の各レジスタとデイレクトリ部

命令番号④：1003 番地の命令「ADD R3, 4100, R2」を実行する。この命令も R2 を指標レジスタとして利用している。実効アドレスは 4100 と R2 の内容を加算した値 4199 番地となり、参照ブロック番号は 42 である。処理の内容は、「B[99] の内容を R3 に加算する」ことになるため、R3 の内容は、P1 = A[99] + B[99] となる。新たなブロック番号 42 が参照されるので、デイレクトリ部 3 に順位を 3 として追加される。これらをまとめると、レジスタ及びデイレクトリ部は次のようになる。

R1	R2	R3	ブロック番号	順位	フラグ
－1	99	P1	デイレクトリ部 1	45	1
			デイレクトリ部 2	41	2
			デイレクトリ部 3	42	3

図 3-4 1003 番地の ADD 命令を実行した直後の各レジスタとデイレクトリ部

問1 キャッシュメモリ（ハードウェア）
(H22春・FE 午後問1)

【解答】

設問 1]	a—カ, b—エ, c—ウ
設問 2]	d—イ, e—ク, f—ウ

【解説】

キャッシュメモリに関するハードウェアの問題である。テーマとしては、ページングアルゴリズムで重要な FIFO (First In First Out) 及び LRU (Least Recently Used) の内容を命令の実行と絡めて問うものである。

対策として、その考え方を学習していた受験者は多いと推測するが、問題文が長く、解答内容の理解に時間を要すること、命令の理解と実行内容を確認しながらのトレース (追跡) が必要であることから、制限時間内に解くという点では難易度は高いといえる。

設問1は映画、確認が比較的容易にできるはずであるので、ここで基礎点を獲得しておきたい。設問2は命令の理解と実行内容を確認しながらのトレースにやや時間がかかるが、落ち着いて解き、ミスをしないう気を付ける必要がある。

キヤッシュを構成するデイトレトリ部、データ部の内容は問題文の図2に示されているが、デイトレトリ部のブロック番号、順位、フラグの意味を正確に把握することが解決のポイントである。

問題文の表 1 で示されているプログラム領域 (命令) とデータ領域の内容、及びプログラムの表 1 で示されているプログラムの実行に伴う段階的な値を P1~P3 と上命令番号①~⑨を付与しており、プログラムの実行に伴う段階的な値を示している。

1. 命令とプログラム領域の内容

表A プログラム領域の内容

命令番号	番地	命令	処理の内容	参照 フロッグ
①	1000	LOAD R1, 4400	—1をR1に設定	45
②	1001	LOAD R2, 4401	iの初期値(99)をR2に設定	45
③	1002	LOAD R3, 4000, R2	A[i]の内容をR3に設定	41
④	1003	ADD R3, 4100, R2	B[i]の内容をR3に加算	42
⑤	1004	ADD R3, 4200, R2	C[i]の内容をR3に加算	43
⑥	1005	STORE R3, 4000, R2	R3の内容をA[i]に格納	41
⑦	1006	STORE R3, 4300, R2	R3の内容をD[i]に格納	44
⑧	1007	ADDR R2, R1	$i \leftarrow i - 1$	—
⑨	1008	JNM 1002	$i \geq 0$ ならば1002番地にジャンプ	—

2. データ領域の主記憶への割付け内容 (各配列領域の添字は 0~99 が対応する)

領域の添字 1 → 0 1 ... 98 99

番地 →

A の配列領域 (アドレス番号 41)

A[0]	A[1]	...	A[98]	A[99]
------	------	-----	-------	-------

番地 →

B の配列領域 (アドレス番号 42)

B[0]	B[1]	...	B[98]	B[99]
------	------	-----	-------	-------

番地 →

C の配列領域 (アドレス番号 43)

C[0]	C[1]	...	C[98]	C[99]
------	------	-----	-------	-------

番地 →

D の配列領域 (アドレス番号 44)

D[0]	D[1]	...	D[98]	D[99]
------	------	-----	-------	-------

番地 →

1 と 99 の格納領域 (アドレス番号 45)

-1	99
----	----

3. プログラムが行う処理内容

$i: 99, i \geq 0, --1$...	处理 1
$\cdot A[i] \leftarrow A[i] + B[i] + C[i]$
$\cdot D[i] \leftarrow A[i]$...	处理 2

疑以言語で表現されているが、処理は添字の初期値を 99 として、 -1 しながら配列領域 $A \sim C$ の要素を加算していく内容である。加算した値は、配列領域 A と D に格納されていく。

iの初期値 99 の場合における処理 1, 2 の段階別内容

- (1) $p_1 \leftarrow A[i] + B[i] = A[99] + B[99]$

```
(2) P2 ← P1 + C[i] = A[i] + B[i] + C[i]
```

$$= A[99] + B[99] + C[99]$$

[設問1]

・空欄 a~c: 命令の実行に伴いデインクトリ部の内容がどうなるかは、設問(1)~(6)に記述されているので、よく把握しながら解けばよい。なお、キヤッシュ更新ロジックは FIFO (最も古くから存在するブロッックと置き換える) であり、置き換えが発生した場合は、順位 1 のデインクトリ部と置き換える。順位の繰下げは -1 すればよい。

表Aに示した内容に基づき、トレースを行うと次のようになる。

命令番号①：1000番地の命令「LOAD R1,4400」を実行する。参照ブロック番号は45で、処理の内容は、「-1をR1に設定する」ものである。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（設問1(6)）こと、及び設問1(1)、(2)からデイレクトリ部の内容は図3-1のようになる。

R1	R2	R3	プロック番号	順位	フラグ
-1			45	1	0
			0	0	0
			0	0	0

順位は最も古くから存在することになり 1

図3-1 1000番地のLOAD命令を実行した直後の各レジスタとデイレクトリ部

したがって、空欄 a は (力) が正解である。

次に、1006 番地の STORE 命令を最初にした直後のダイレクトリ部の内容を吟味してみる。以下、順にその様子を示す。

命令番号②: 1001番地の命令「LOAD R2, 4401」を実行する。参照ブロック

番号は 45 で、処理の内容は、「1 の初期値(99)を R2 に設定する」ものである。これは命令番号①とはほぼ等機能の内容で、参照ブロック番号 45 は既にキャッシュ内に存在するので、ダイレクトリ部の内容に変化がない。なお、フラグは、対応するデータ部のバッファに保存されている内容が CPU の処理によって変更された場合 (STORE 命令実行時) に 1 と変化するが、ここでは変更は発生しないので 0 のままであり、R2 に 99 が設定される。

	R1	R2	R3		プロック番号	順位	フアラジ
—1	99			デイレクトリ部1	45	1	0
				デイレクトリ部2	0	0	0
				デイレクトリ部3	0	0	0

図3-2 1001番地のLOAD命令を実行した直後の各レジスタとディレクトリ部

命令番号③：1002番地の命令「LOAD R3,4000,R2」を実行する。この命令はR2を指標レジスタとして利用している。実効アドレスは4000とR2の内容を加算した値が示す番地となるので、 $4000 + 99(R2 \text{ の内容}) = 4099$ 番地である。処理の内容は、「[A11 = 99]の内容をR3に設定する」ことで、参照ブロック番号は41になる。新たなブロックが参照されることになるため、デイレクトリ部2に順位を2として追加される。これらをまとめると、レジスタ及びデイレクトリ部は次のようになる。

R1	R2	R3		ズロツク番号	順位	フラゲ
-1	99	A[99]	ヱィレクトリ部1	45	1	0
			ヱィレクトリ部2	41	2	0
			ヱィレクトリ部3	0	0	0

図3-3 1002番地のLOAD命令を実行した直後の各レジスタとディスプレイ部

命令番号④：1003 番地の命令「ADD R3,4100,R2」を実行する。この命令も R2 を指標レジスタとして利用している。実効アドレスは 4100 と R2 の内容を加算した値 4199 番地となり、参照ブロック番号は 42 である。処理の内容は、「B[199]の内容を R3 に加算する」ことになるため、R3 の内容は、 $P1 = A[99] + B[99]$ となる。新たなブロック番号 42 が参照されるので、デイレクトリ部 3 に順位を 3 として追加される。これらをまとめると、レジスタ及びデイレクトリ部は次のようになる。

	R1	R2	R3		プロック番号	順位	ララゲ
—1	99	P1		デイレクトリ部1	45	1	0
				デイレクトリ部2	41	2	0
				デイレクトリ部3	42	3	0

図3-4 1003番地のADD命令を実行した直後の各レジスタとデイレクトリ部