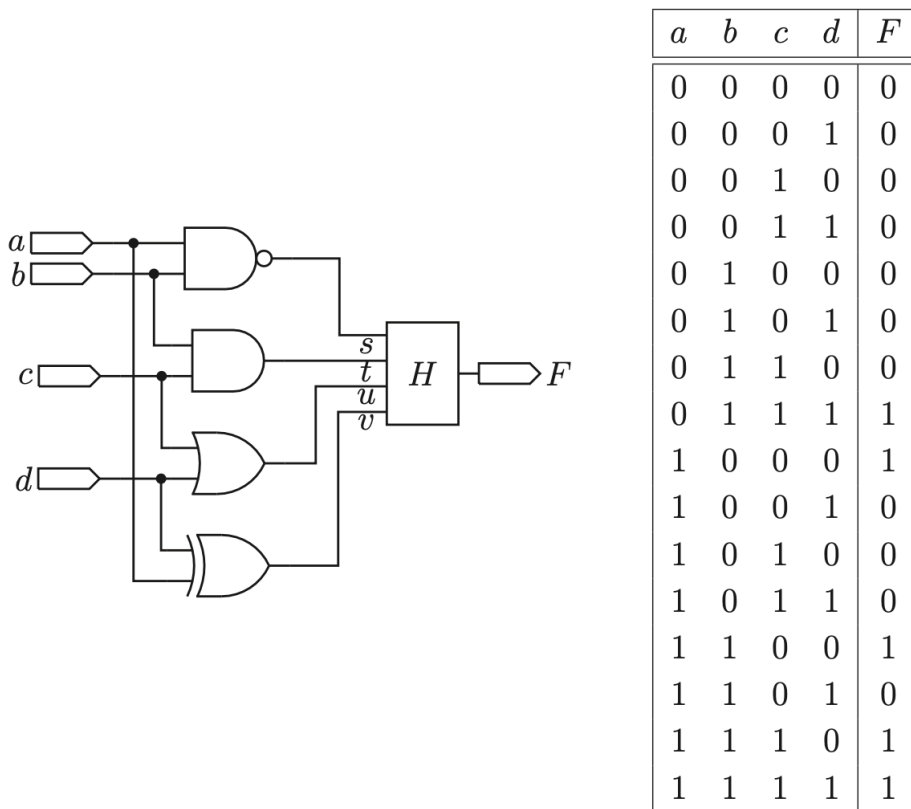


## 令和五年

【問 1】 下図の様にいくつかの論理ゲートと部分回路  $H$  から構成される論理回路の出力の論理関数  $F(a, b, c, d)$  が以下の様な真理値表で表される時, 部分回路  $H$  の論理関数  $H(s, t, u, v)$  の最簡積和形を示せ. ただし, 論理関数の最簡積和形とはその論理関数を表す積和形論理式のうち, 積項数が最小のものを指す. 積項数が等しい積和形論理式が複数ある場合にはそのなかでリテラル数が最小のものを指す.



解答

$$s = \overline{ab} \quad t = bc \quad u = c + d \quad v = a \oplus d \quad (1)$$

<b>a</b>	<b>b</b>	<b>c</b>	<b>d</b>	<b>s</b>	<b>t</b>	<b>u</b>	<b>v</b>	<b>F</b>
0	0	0	0	1	0	0	0	0
0	0	0	1	1	0	1	1	0
0	0	1	0	1	0	1	0	0
0	0	1	1	1	0	1	1	0
0	1	0	0	1	0	0	0	0
0	1	0	1	1	0	1	1	0
0	1	1	0	1	1	1	0	0
0	1	1	1	1	1	1	1	1
1	0	0	0	1	0	0	1	1
1	0	0	1	1	0	1	0	0
1	0	1	0	1	0	1	1	0
1	0	1	1	1	0	1	0	0
1	1	0	0	0	0	0	1	1
1	1	0	1	0	0	1	0	0
1	1	1	0	0	1	1	1	1
1	1	1	1	0	1	1	0	1

<b>s</b>	<b>t</b>	<b>u</b>	<b>v</b>	<b>F</b>
0	0	0	0	x
0	0	0	1	1
0	0	1	0	0
0	0	1	1	x
0	1	0	0	x
0	1	0	1	x
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	x
1	1	0	1	x
1	1	1	0	0
1	1	1	1	1

<b>st\uv</b>	<b>00</b>	<b>01</b>	<b>11</b>	<b>10</b>
<b>00</b>	x	1	x	0
<b>01</b>	x	x	1	1
<b>11</b>	x	x	1	0
<b>10</b>	0	1	0	0

st\uv	00	01	11	10
00	x	1	x	0
01	x	x	1	1
11	x	x	1	0
10	0	1	0	0

$$H = \bar{s}t + \bar{u}v + tv \quad (2)$$

【問2】パイプライン式データパスを有するマイクロプロセッサについて考える。以下の式で示すように、プログラム実行時間  $ET$  は、プログラム実行のために処理される総命令数  $IC$ 、クロックサイクル当り実行命令数  $IPC$ 、ならびに、動作周波数  $F$ 、の3つのパラメータを用いて表現できる。

$$ET = \frac{IC}{IPC \times F}$$

以下の各問いに答えよ。

- (1) 命令発行幅が1のインオーダマイクロプロセッサを考える。シングルサイクル・データパス方式（1命令の実行を1クロックサイクルで処理する方式）と比較した場合、パイプライン式データパスの実装が  $IC$ 、 $IPC$ 、 $F$  に与える影響をそれぞれ説明せよ。なお、各パラメータにおいて影響がない場合は「影響なし」と答えること。
- (2) このパイプライン式データパスにおいて命令発行幅を2へ増加し、インオーダ命令実行のスーパスカラ方式へと拡張した（依存関係のない命令を最大で2個同時に実行できる）。この拡張が  $IC$ 、 $IPC$ 、 $F$  に与える影響をそれぞれ説明せよ。なお、各パラメータにおいて影響がない場合は「影響なし」と答えること。
- (3) 命令発行幅は4と仮定する。このパイプラインで達成できる  $IPC$  の上限を答えよ。

## 解答

(1)

$IC$  : 影響なし  $IPC$  : 低下する可能性がある  $F$  : より高い (3)

(2)

$IC$  : 影響なし  $IPC$  : より高い  $F$  : より低い (4)

(3)

$IPC_{max} = 4$  (5)

**【問3】** コンピュータのメモリシステムについて考える。マイクロプロセッサにダイレクトマップ・キャッシュが搭載されているものとする。ワードアドレッシング方式を採用しており、ワードサイズは4バイト、キャッシュ・サイズは16バイト、ブロックサイズは4バイト、アドレス長は4ビットである。キャッシュの初期状態は空であったが、これまでに  $1101 \Rightarrow 1010 \Rightarrow 1111 \Rightarrow 1101$  のワードアドレス (2進表現) に対してメモリアクセスが順次発生している。このとき、メモリアクセス①～⑤が順次発生したとする。

メモリアクセス	ワードアドレス (2進表現)
①	1010
②	1001
③	1000
④	0011
⑤	1111

以下の各問いに答えよ。

- (1) メモリアクセス①～⑤のうち、キャッシュ・ヒットとなるメモリアクセスをすべて答えよ。該当するメモリアクセスがない場合は「該当なし」と答えること。
- (2) メモリアクセス①～⑤のうち、初期参照ミスとなるメモリアクセスをすべて答えよ。該当するメモリアクセスがない場合は「該当なし」と答えること。
- (3) 競合ミスを削減するためにはこのキャッシュをどのように改良すれば良いか説明せよ。また、改良によるデメリットがあればあわせて説明せよ。

## 解答

(1)

① (6)

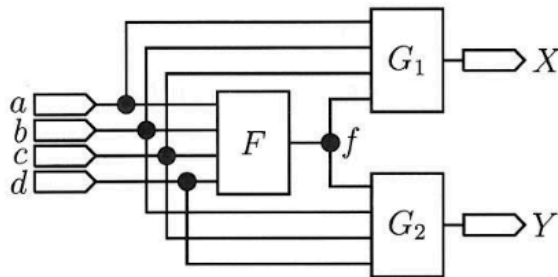
(2)

③ (7)

(3)

令和四年

【問 1】 下図の論理回路の出力  $X, Y$  および内部回路  $G_1, G_2$  の論理関数が以下の様な真理値表で表されるとき、部分回路  $F$  の論理関数  $F(a, b, c, d)$  の最簡積和形を示せ。ただし、論理関数の最簡積和形とは、その論理関数を表す積和形論理式のうち、積項数が最小のものを指す。積項数が等しい積和形論理式が複数ある場合には、そのなかでリテラル数が最小のものを指す。



$a$	$b$	$c$	$d$	$X$	$Y$
0	0	0	0	0	1
0	0	0	1	0	0
0	0	1	0	0	1
0	0	1	1	0	0
0	1	0	0	0	1
0	1	0	1	0	1
0	1	1	0	0	1
0	1	1	1	0	0
1	0	0	0	1	1
1	0	0	1	0	0
1	0	1	0	0	1
1	0	1	1	0	0
1	1	0	0	1	1
1	1	0	1	1	0
1	1	1	0	1	0
1	1	1	1	0	0

$a$	$b$	$c$	$f$	$G_1$
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

$f$	$b$	$c$	$d$	$G_2$
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

## 解答

<b>a</b>	<b>b</b>	<b>c</b>	<b>d</b>	<b>f</b>
0	0	0	0	x
0	0	0	1	0
0	0	1	0	x
0	0	1	1	x
0	1	0	0	x
0	1	0	1	1
0	1	1	0	1
0	1	1	1	x
1	0	0	0	1
1	0	0	1	0
1	0	1	0	x
1	0	1	1	x
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

<b>ab\cd</b>	<b>00</b>	<b>01</b>	<b>11</b>	<b>10</b>
<b>00</b>	x		x	x
<b>01</b>	x	1	x	1
<b>11</b>			1	
<b>10</b>	1		x	x

$$F = \bar{a}b + cd + \bar{b}\bar{d}$$

(9)



**【問 2】** 5つのステージからなるパイプライン型データパスを有するインオーダー・マイクロプロセッサについて考える。実装されたパイプラインステージは、IF（命令取得）、ID（命令デコード）、EX（実行）、MEM（メモリアクセス）、ならびに、WB（ライトバック）である。加算命令の実行における各ステージの処理内容は以下の表に従う。ここで、各パイプラインステージの実行は常に1クロックサイクルで完了できると仮定する。また、WBステージでレジスタに書き込まれた値は、同一クロックサイクルにて、後続命令のIDステージで読み出し可能である。さらに、全ての RAW（Read-After-Write）ハザードはパイプラインストールにより解決する。以下の各問に答えよ。

パイプライン式データパスの動作

ステージ	加算命令 (add \$x, \$y, \$z) 実行における各ステージの処理内容
IF	メモリより実行すべき命令を取得し、次命令取得のためにプログラム・カウンタを更新
ID	命令の解読。レジスタファイルからレジスタ \$y ならびに \$z を読み出し。
EX	レジスタファイルから読み出した \$y と \$z の内容を加算。
MEM	特に無し（加算結果を WB ステージへ転送）。
WB	加算結果をレジスタファイル内のレジスタ \$x に書き込み。

(1) 以下に示すアセンブリプログラムについて考える。各行において‘#’記号から右はコメントである。プログラム中に存在するフロー依存関係について、どの命令が、どの命令のどのレジスタに関して依存しているかをすべて列挙せよ。

```

add $1, $3, $5 # <1>
add $9, $2, $3 # <2>
add $6, $3, $3 # <3>
add $3, $4, $3 # <4>
add $4, $7, $1 # <5>
add $5, $7, $4 # <6>
add $9, $3, $6 # <7>
add $2, $7, $6 # <8>

```

(2) 命令発行幅は1と仮定する。このアセンブリプログラムの実行に要するクロックサイクル数を答えよ。

(3) 命令パイプラインの命令発行幅を2へ増加し、インオーダー・スーパースカラ方式へと拡張する（依存関係のない命令を最大で2個同時に実行できる）。拡張後の命令パイプラインにて、このアセンブリプログラムの実行に要するクロックサイクル数を答えよ。

(4) 上記 (3) の拡張によりクロック周波数が 5 % 低下した。この拡張による性能向上率を答えよ。

**解答**

(1)

$$1 \Rightarrow 5 (\$1), 3 \Rightarrow 7 (\$6), 3 \Rightarrow 8 (\$6), 4 \Rightarrow 7 (\$3), 5 \Rightarrow 6 (\$4) \quad (10)$$

(2)

$$14 \quad (11)$$

(3)

$$12 \quad (12)$$

(4)

$$1.11 \quad (13)$$

**【問3】** キャッシュメモリにおける「初期参照ミス」「競合性ミス」「容量性ミス」とは何かそれぞれ説明せよ。

**解答**

初期参照ミス：キャッシュラインを最初にアクセスするときに起こるミス

競合性ミス：同じインデックスをもつ異なるキャッシュラインにアクセスすることで起こるミス (14)

容量性ミス：キャッシュしたいライン数がキャッシュ容量を上回ることで起こるミス

**令和三年**

【問 1】 与えられた真理値表で表される論理関数  $F(a, b, c, d)$  が以下の論理式を満たす時、論理関数  $G(X, Y, Z, W)$  の最簡積和形を示せ。ただし、最簡積和形とは積和形論理式のうち、積項数が最小のものを指す。積項数が等しい積和形論理式が複数ある場合にはそのなかでリテラル数が最小のものを指す。

$a$	$b$	$c$	$d$	$F$
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

$$X = \bar{a}\bar{b}\bar{c} + \bar{b}d + ab\bar{d}$$

$$Y = \bar{a}c\bar{d} + bcd + \bar{a}\bar{b}d + a\bar{c}\bar{d}$$

$$Z = \bar{a}b\bar{c} + b\bar{c}d + \bar{b}d$$

$$W = \bar{a}b\bar{d} + abd + cd + \bar{a}c$$

$$F(a, b, c, d) = G(X, Y, Z, W)$$

解答

<b>a</b>	<b>b</b>	<b>c</b>	<b>d</b>	<b>X</b>	<b>Y</b>	<b>Z</b>	<b>W</b>	<b>F</b>
0	0	0	0	1	0	0	0	0
0	0	0	1	1	0	1	0	0
0	0	1	0	0	1	0	1	0
0	0	1	1	1	0	1	1	1
0	1	0	0	0	0	1	1	0
0	1	0	1	0	0	1	0	1
0	1	1	0	0	1	0	1	0
0	1	1	1	0	1	0	1	0
1	0	0	0	0	1	0	0	1
1	0	0	1	1	1	1	0	1
1	0	1	0	0	0	0	0	1
1	0	1	1	1	1	1	1	0
1	1	0	0	1	1	0	0	1
1	1	0	1	0	0	1	1	0
1	1	1	0	1	0	0	0	0
1	1	1	1	0	1	0	1	0

<b>X</b>	<b>Y</b>	<b>Z</b>	<b>W</b>	<b>G</b>
0	0	0	0	1
0	0	0	1	x
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	x
0	1	1	1	x
1	0	0	0	0
1	0	0	1	x
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	x
1	1	1	0	1
1	1	1	1	0

<b>XYZW</b>	<b>00</b>	<b>01</b>	<b>11</b>	<b>10</b>
<b>00</b>	1	x		1
<b>01</b>	1		x	x
<b>11</b>	1	x		1
<b>10</b>		x	1	

<b>XY\ZW</b>	<b>00</b>	<b>01</b>	<b>11</b>	<b>10</b>
<b>00</b>	1	x		1
<b>01</b>	1		x	x
<b>11</b>	1	x		1
<b>10</b>		x	1	

$$G = \bar{X}\bar{W} + Y\bar{W} + XYW \quad (15)$$

**【問2】** 5つのステージからなるパイプライン式データパスを有するマイクロプロセッサについて考える。実装されたパイプラインステージは、IF（命令取得）、ID（命令デコード）、EX（実行）、MEM（メモリアクセス）、ならびに、WB（ライトバック）である。以下の各問いに答えよ。

- (1) IF, ID, EX, MEM, WBの遅延時間は、それぞれ、240 ps, 400 ps, 200 ps, 250 ps, 180 psである。このデータパスの最大動作周波数を答えよ（単位は GHz）。
- (2) IF, ID, EX, MEM, WBのいずれか1つを2つのステージに分割し、パイプラインステージ数を5から6へと増加することを考える。ここで、分割された各パイプラインステージの遅延時間は、分割前のパイプラインステージの遅延時間の半分になると仮定する。パイプラインの動作周波数を最大にするために分割すべきパイプラインステージを選択せよ。また、この設計最適化により達成できる最大動作周波数を答えよ（単位は GHz）。
- (3) あるプログラムの実行において、上記(2)のパイプラインステージの分割によりCPI（Clock cycles Per Instruction）が10%増加した。ここで、パイプラインステージ分割はCPIの増加以外の悪影響は生じないと仮定する。このパイプラインステージ分割によって得られた性能向上比を答えよ。
- (4) パイプライン段数を増加することの利点と欠点を説明せよ。

## 解答

(1)

$$F_1 = \frac{1}{400 \times 10^{-12}} \text{Hz} = 2.5 \text{GHz} \quad (16)$$

(2)

$$F_2 = \frac{1}{250 \times 10^{-12}} \text{Hz} = 4 \text{GHz} \quad (17)$$

(3)

$$1.45 \quad (18)$$

(4)

利点:

1. クロック周波数の向上: パイプライン段数を増やすことで、各段の論理ゲート数が減り、クロックサイクルタイムが短くなります。その結果、クロック周波数が向上し、プロセッサの性能が向上する可能性があります。
2. 並列処理の向上: 各段が独立して動作するため、複数の命令が同時に処理されることができ、これにより、高い命令実行レートが達成され、システム全体のスループットが向上します。
3. 効率的なリソース利用: パイプライン段数が増えると、各リソース (ALU、レジスタファイル、キャッシュなど) がより効率的に使用される可能性があります。これにより、ハードウェアの消費電力やエネルギー効率が向上することがあります。

欠点:

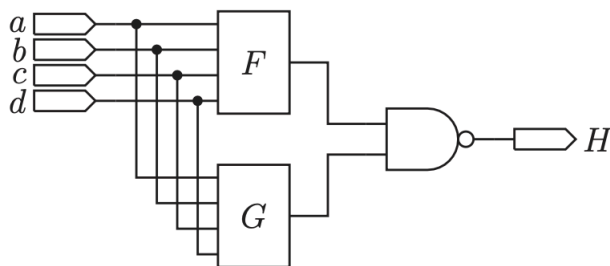
1. パイプラインハザード: パイプライン段数が増えると、データハザード、制御ハザード、構造ハザードなどのパイプラインハザードが発生しやすくなります。これらのハザードは、パイプラインの効率を低下させる可能性があります。
2. 複雑さの増加: パイプライン段数が増えると、回路設計や制御ロジックの複雑さが増します。これにより、設計や検証のコストが増加し、バグの発生リスクが高まる可能性があります。
3. レイテンシの増加: パイプライン段数が増えると、各命令の完了にかかる時間 (レイテンシ) が長くなる可能性があります。これは、特にレイテンシに敏感なアプリケーションで問題となることがあります。

**【問3】** キャッシュタグ・フィールド, キャッシュインデックス・フィールド, キャッシュブロックオフセット・フィールドからなる 32 ビットのメモリアドレスを入力とするダイレクトマップ・キャッシュメモリについて考える。バイトアドレッシング方式を採用しており、1語は4バイト, キャッシュサイズは16キロバイト, キャッシュブロックサイズは32バイトとする。キャッシュタグ・フィールドのビット幅を答えよ。

令和二年

【問1】 論理関数  $H(a, b, c, d)$  を図で示されるように関数  $F(a, b, c, d)$ ,  $G(a, b, c, d)$  および NAND ゲートを使って実現することを考える. 関数  $H$  および関数  $G$  の真理値表が以下のように与えられる時, 関数  $F$  の最簡積和形を示せ.

$a$	$b$	$c$	$d$	$H$	$G$
0	0	0	0	0	1
0	0	0	1	1	1
0	0	1	0	1	0
0	0	1	1	1	1
0	1	0	0	1	0
0	1	0	1	1	1
0	1	1	0	1	0
0	1	1	1	1	0
1	0	0	0	1	0
1	0	0	1	1	1
1	0	1	0	0	1
1	0	1	1	0	1
1	1	0	0	1	1
1	1	0	1	1	1
1	1	1	0	1	0
1	1	1	1	1	0



解答

$$H = \overline{FG} \tag{21}$$



<b>a</b>	<b>b</b>	<b>c</b>	<b>d</b>	<b>F</b>
0	0	0	0	1
0	0	0	1	0
0	0	1	0	x
0	0	1	1	0
0	1	0	0	x
0	1	0	1	0
0	1	1	0	x
0	1	1	1	x
1	0	0	0	x
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	x
1	1	1	1	x

<b>ab\cd</b>	<b>00</b>	<b>01</b>	<b>11</b>	<b>10</b>
<b>00</b>	1			x
<b>01</b>	x		x	x
<b>11</b>			x	x
<b>10</b>	x		1	1

ab\cd	00	01	11	10
00	1			X
01	X		X	X
11			X	X
10		X	1	1

**【問 2】** 5つのステージからなるパイプライン式データパスを有するマイクロプロセッサについて考える。実装されたパイプラインステージは、IF（命令取得）、ID（命令デコード）、EX（実行）、MEM（メモリアクセス）、ならびに、WB（ライトバック）である。以下の各問いに答えよ。

- (1) プログラム実行時間は、「実行命令数」「平均 CPI(Clock cycles Per Instruction)」「クロックサイクル時間」の積で近似できる。シングルサイクル・データパス方式（1命令の実行を1クロックサイクルで実行する方式）と比較した場合、パイプライン処理がプログラム実行時間に与える影響を説明せよ。
- (2) パイプライン式データパスで発生する RAW (Read After Write) ハザードを解消する代表的な実装として「パイプラインストール」と「データフォワーディング」がある。これらの実装法の利点と欠点を説明せよ。
- (3) 命令発行幅は1と仮定する。このパイプラインで達成できる IPC (Instructions Per Clock cycle) の上限を答えよ。

## 解答

(1)

1. クロックサイクル時間の短縮: パイプライン処理では、各ステージが短いクロックサイクル時間で実行できるため、全体のクロックサイクル時間が短くなります。
2. 並行性の向上: パイプライン処理では、複数の命令が同時に実行されるため、全体の実行速度が向上します。
3. パイプラインハザードの影響: パイプライン処理では、パイプラインハザードが発生することでCPIが増加し、実行時間が長くなる可能性があります。

(2)

1. パイプラインストール:

利点:

- シンプルな実装で、ハードウェアの複雑さが比較的少ない。
- データ依存性が発生した場合に、後続の命令を一時停止してデータが利用可能になるのを待つため、ハザードが解消される。

欠点:

- 後続の命令がストールされるため、パイプラインの効率が低下し、IPC (Instructions Per Cycle) が減少する可能性がある。
- パイプラインストールが頻繁に発生すると、プロセッサの性能が大幅に低下する。

1. データフォワーディング:

利点:

- 後続の命令がデータを待つことなく実行できるため、パイプラインの効率が向上し、IPCが向上する可能性がある。
- データ依存性がある場合でも、パイプラインストールが発生しにくくなるため、プロセッサの性能が向上する。

欠点:

- ハードウェアの複雑さが増すため、実装が難しくなる。
- すべてのRAWハザードがデータフォワーディングで解消できるわけではなく、場合によってはパイプラインストールが必要になることがある (例: ロード命令の結果がすぐに利用可能でない場合)。

(3)

1

(22)

**【問3】** 16ビットのアドレス ( $\text{adr}[15:0]$ ) を入力とする 2 ウェイ・セットアソシアティブキャッシュの設計について考える。バイトアドレッシング方式であり 1 語は 4 バイトとする。各キャッシュアクセスにおいて、 $\text{adr}[15:8]$ 、 $\text{adr}[7:4]$  ならびに  $\text{adr}[3:0]$  は、それぞれ、タグフィールド、インデックスフィールド、オフセットフィールドとして参照される。以下の問いに答えよ。

(1) キャッシュブロックサイズを答えよ。

(2) キャッシュサイズを答えよ。

(3) レベル1 キャッシュと主記憶からなるメモリシステムを想定した場合、平均メモリアクセス時間 (AMAT: Average Memory Access Time) は下の式で表される。ここで、キャッシュヒット時間はキャッシュアクセス時間と同義と考えて良い。より高いキャッシュ連想度を採用することの利点と欠点を AMAT を用いて説明せよ。

$$\text{AMAT} = \text{キャッシュヒット時間} + \text{キャッシュミス率} \times \text{キャッシュミスペナルティ}$$

## 解答

(1)

16バイト

(23)

(2)

512バイト

(24)

(3)

利点:

1. キャッシュミス率の低下: キャッシュ連想度が高いと、キャッシュ内でのデータの配置が柔軟になるため、キャッシュミスが発生しにくくなります。これにより、キャッシュミス率が低下し、AMATが短くなります。
2. ローカリティの向上: キャッシュ連想度が高いと、データのローカリティが向上し、特定のデータに対するアクセスが高速化されます。これもAMATを短くする要因となります。

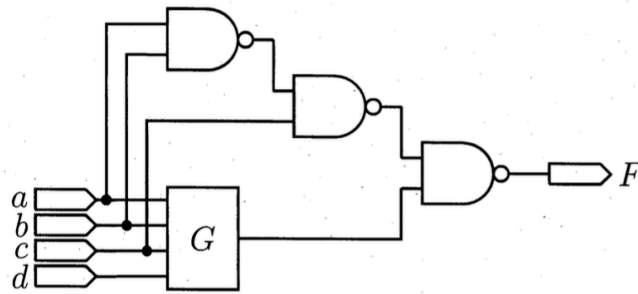
欠点:

1. キャッシュヒット時間の増加: キャッシュ連想度が高いと、キャッシュ内でのデータの検索に時間がかかる可能性があります。これにより、キャッシュヒット時間が増加し、AMATが長くなることがあります。
2. ハードウェアの複雑さとコスト: 高いキャッシュ連想度を実現するには、より複雑なハードウェアが必要となり、コストが増加します。このコスト増加は、性能向上のためのトレードオフとして考慮されるべきです。

令和元年

【問1】以下の真理値表で与えられた論理関数  $F(a, b, c, d)$  を図で示されるように関数  $G(a, b, c, d)$  および NAND ゲートを使って実現することを考える。関数  $G$  の最簡積和形を示せ。

$a$	$b$	$c$	$d$	$F$
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1



解答

$$F = \overline{abcG}$$

(25)

<b>a</b>	<b>b</b>	<b>c</b>	<b>d</b>	<b>G</b>
0	0	0	0	1
0	0	0	1	1
0	0	1	0	x
0	0	1	1	x
0	1	0	0	1
0	1	0	1	0
0	1	1	0	x
0	1	1	1	x
1	0	0	0	1
1	0	0	1	1
1	0	1	0	x
1	0	1	1	x
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

<b>ab\cd</b>	<b>00</b>	<b>01</b>	<b>11</b>	<b>10</b>
<b>00</b>	1	1	x	x
<b>01</b>	1		x	x
<b>11</b>	1			1
<b>10</b>	1	1	x	x

ab\cd	00	01	11	10
00	1	1	x	x
01	1		x	x
11	1			1
10	1	1	x	x

$$G = \bar{b} + \bar{d} \quad (26)$$

【問2】5つのステージからなるパイプライン式データパスを有するマイクロプロセッサについて考える。実装されたパイプラインステージは、IF（命令取得）、ID（命令デコード）、EX（実行）、MEM（メモリアクセス）、ならびに、WB（ライトバック）である。以下の各問いに答えよ。

- (1) パイプラインの導入によりプロセッサ性能が向上する理由を説明せよ。
- (2) このパイプライン構造で発生する RAW (Read After Write) ハザードを解消する方法を少なくとも2つ挙げ、それぞれの実現法を簡潔に説明せよ。
- (3) プログラム実行時間は、「実行命令数」「CPI(Clock cycles Per Instruction)」「クロックサイクル時間」の積で近似できる。このパイプラインのステージ数を増加した場合、改善を期待できる項を選択し、その理由を説明せよ。
- (4) 一般に、パイプライン段数を増加し続けた場合、プロセッサ性能向上の度合いは次第に小さくなる傾向にある。その理由を述べよ。

## 解答

(1)

1. 並行性の向上: パイプライン技術を使用することで、各ステージが同時に異なる命令を処理できます。これにより、プロセッサは複数の命令を同時に実行できるようになり、全体的なスループットが向上します。
2. タスクの分割: パイプラインは、命令の実行をいくつかのステージに分割します。これにより、各ステージは特定のタスクに特化し、そのタスクを効率的に実行できます。これにより、命令の実行時間が短縮され、性能が向上します。

3. クロックサイクルの短縮: 各ステージが独立して動作するため、クロックサイクルは各ステージの最も遅い部分に合わせる必要があります。これにより、クロックサイクルが短くなり、プロセッサの性能が向上します。

(2)

1. ストール (Stall) :

ストールは、RAWハザードが解消されるまで次の命令の実行を一時停止する方法です。これは、データが前の命令から利用可能になるまで待つことで、RAWハザードを回避します。ただし、ストールはプロセッサの性能に悪影響を与える可能性があります。

実現方法: ハードウェアは、RAWハザードが検出された場合、次の命令の実行を一時停止し、前の命令が書き込みを完了するのを待ちます。書き込みが完了したら、次の命令が再開され、ハザードが解消されます。

2. フォワードイング (Forwarding) またはバイパス (Bypass) :

フォワードイングは、前の命令が書き込むデータを、次の命令がそれを読む前に直接転送する方法です。これにより、次の命令が待つことなくデータを読むことができ、RAWハザードが解消されます。

実現方法: ハードウェアは、前の命令が書き込むデータを、次の命令がそれを必要とするステージに直接送信します。これにより、次の命令はデータを待たずに読むことができ、RAWハザードが解消されます。

(3)

パイプラインのステージ数を増加させた場合、改善を期待できる項目は、「CPI (Clock cycles Per Instruction) 」および/または「クロックサイクル時間」です。

パイプラインのステージ数を増やすことで、各ステージで行う作業が少なくなり、クロックサイクル時間が短くなる可能性があります。これにより、クロック周波数が速くなります。さらに、より多くのパイプラインステージが、より良い命令レベルの並列性につながり、平均CPIが低下する可能性があります。ただし、この改善は次第に減少する利益によって制限されます。

(4)

1. 分岐予測の誤り率が増加する:

パイプラインが深くなると、分岐予測が誤る可能性が増加し、パイプラインがしばしばフラッシュされることとなります。これにより、性能向上が抑制されることがあります。

2. パイプラインのバランスが悪化する:

パイプライン段数が増加すると、各ステージの処理時間が異なる場合、最も遅いステージに従属することがあります。これにより、全体の性能が低下する可能性があります。

3. プロセッサの複雑さが増加する:

パイプライン段数が増加すると、プロセッサの設計や実装がより複雑になり、デバッグや最適化が困難になることがあります。また、消費電力やチップ面積も増加する可能性があります。

4. 指令レベルの並列性 (ILP) の限界:

プロセッサが同時に実行できる命令数には自然な限界があります。パイプライン段数を増加させても、その限界を超えることはできません。したがって、性能向上の度合いは次第に小さくなります。



【問3】 16ビットのアドレス (adr[15:0]) を入力とするダイレクトマップキャッシュの設計について考える。バイトアドレッシング方式であり 1語は 4バイトとする。各キャッシュアクセスにおいて、adr[15:8], adr[7:4]ならびに adr[3:0] は、それぞれ、タグフィールド、インデックスフィールド、オフセットフィールドとして参照される。以下の問いに答えよ。

(1) キャッシュブロックサイズを答えよ。

(2) キャッシュブロックの総数を答えよ。

(3) キャッシュの初期状態は空であるとする。以下の 16進表現されたバイトアドレスに対してメモリアクセスが順次発生した場合のキャッシュ・ミス率を答えよ。

0x0000 ⇒ 0x0004 ⇒ 0x0020 ⇒ 0x1120 ⇒ 0x1104 ⇒ 0x0004 ⇒ 0x1120 ⇒ 0x0020  
⇒ 0x0024 ⇒ 0x0020

## 解答

(1)

16バイト (27)

(2)

16 (28)

(3)

0.6 (29)