

計算システム演習略解

(1) `add r3,r1,r2`
`sub r4,r4,r3`

(2) `lw r3,16(r1)`
`add r3,r3,r2`
`sw r3,16(r1)`

(3) 算術命令 : $2 + 1000 \times 2$
 ロード命令 : 1000
 分岐命令 : 1000
 $\Rightarrow 2002 \times 4 + 1000 \times 5 + 1000 \times 3 = 16008$
 $CPI = 16008/4002 = 4$

(4) 算術命令 : $2 + 500 \times 3$
 ロード命令 : 500×2
 分岐命令 : 500
 $\Rightarrow 1502 \times 4 + 1000 \times 5 + 500 \times 3 = 12508$
 $CPI = 12508/3002 = 4.167$

(5) 1cycle は 2ns
 $12508 \times 2[ns] = 25016[ns](25.016[\mu s])$

(6) 総サイクル数は $1502 \times 4 + 1000 \times 4 + 500 \times 3 = 11508$
 総実行時間は $11508 \times 2.5[ns] = 28770$
 $28770/25016=1.15$ 約 15パーセント長くなる

(7) 4byte データへの連続アクセスなので、ラインサイズ 32byte の場合 8 回に 1 回ミスする。従ってヒット率は 87.5%

(8) direct: 7misses

	0(m)	4(m)	0(m)	8(m)	0(m)	12(m)	4(m)
index 0:	0	4	0	8	0	12	4
index 1:							
index 2:							
index 3:							

2-way: 5misses

	0(m)	4(m)	0(h)	8(m)	0(h)	12(m)	4(m)
index 0:	0	0/4	0/4	0/8	0/8	0/12	4/12
index 1:							

full: 4misses

	0(m)	4(m)	0(h)	8(m)	0(h)	12(m)	4(h)
index 0:	0	0/4	0/4	0/4/8	0/4/8	0/4/8/12	0/4/8/12

(9) direct: 7misses

replacement algorithm は関係ない

2-way: 6misses

	0(m)	4(m)	0(h)	8(m)	0(m)	12(m)	4(m)
index 0:	0	0/4	0/4	8/4	8/0	12/0	12/4
index 1:							

full: 4misses

	0(m)	4(m)	0(h)	8(m)	0(h)	12(m)	4(h)
index 0:	0	0/4	0/4	0/4/8	0/4/8	0/4/8/12	0/4/8/12

(10) LRU: 7 page faults

	0(f)	1(f)	2(f)	3(f)	0(f)	3	5(f)	2(f)	3
pages:	0	0/1	0/1/2	3/1/2	3/0/2	3/0/2	3/0/5	3/2/5	3/2/5

FIFO: 8 page faults

	0(f)	1(f)	2(f)	3(f)	0(f)	3	5(f)	2(f)	3(f)
pages:	0	0/1	0/1/2	1/2/3	2/3/0	2/3/0	3/0/5	0/5/2	5/2/3

OPT: 5 page faults

	0(f)	1(f)	2(f)	3(f)	0	3	5(f)	2	3
pages:	0	0/1	0/1/2	0/3/2	0/3/2	0/3/2	5/3/2	5/3/2	5/3/2

(11) $CPI = 0.25 \times 4 + 0.2 \times 3 + 0.3 \times 5 + 0.25 \times 4 = 4.1$

(12) $CPI = 0.25 \times 4 + 0.2 \times 3 + 0.3 \times (0.8 \times 5 + 0.2 \times 25) + 0.25 \times (0.8 \times 4 + 0.2 \times 24) = 6.3$

(13) 改良後の $CPI = 0.25 \times 4 + 0.2 \times 3 + 0.3 \times (0.9 \times 5 + 0.1 \times 25) + 0.25 \times (0.9 \times 4 + 0.1 \times 24) = 5.2$
 改良前の実行時間 / 改良後の実行時間 = $6.3 / (5.2 / 0.95) = 1.15$
 従って、約 15% 向上する。

(14) big endian の場合は MSD (Most Significant Digit) から格納するので 0x12
 little endian の場合は LSD (Least Significant Digit) から格納するので 0x78

$$(15) \text{ availability} = MTBF / (MTBF + MTTR) = 10 \times 60 / (10 \times 60 + 10) = 0.9836 \\ 98.36\%$$