**ברק ברנצויג 313593840**

**ירדן גבאי 318222171**

**שאלה 1 א:**

ADDI R1, R0, 24

ADDI R2, R0, 60 R2=60

ADDI R3, R0,90 R3=90

ADD R13, R0, R0 R13=0

ADD R12, R0, R0 R12=0

ADD R6, R0, R0 R6=0

LOOP: LW R7, 1000(R13)

SUBI R1, R1, 1

SLT R10, R7, R2

BEQ R10, R0, LABEL1 (CB)

DS אין אפשרות למלא את הדילאי סלוט הזה

JUMP LABEL2

ADDI R12, R12, 1

LABEL1: SLT R10, R7, R3

BNE R10, R0, LABEL2(CB)

ADDI R16, R16, 1

LABEL2:

BNE R1, R0, LOOP (NCB)

ADDI R13, R13, 4

**1.ב**

ADDI R1, R0,1000

ADDI R10, R0, 2000

ADDI R2, R0, 10

LOOP: LW R3, 0(R1)

SUBI R2, R2,1

ANDI R4, R3, 3 תלות מטופלת עי"י הFU

BEQ R4, 40, LABEL1

ADDI R1, R1, 4

JUMP LABEL2

SW R3, 0(R10)

LABEL1: SW R0, 0(R10)

LABEL2:

BNE R2, R0, LOOP

ADDI R10, R10, 4

**1.ג**

ADDI R2, R0, 20

ADDI R3, R0, 52

LOOP: SUBI R2, R2, 4

FU BNE R2, R0, LOOP

LW R1,16(R2)

SW R1, 0(R3) SAL

ADDI R3, R3, 4

**2.א**

ADDI R1, R0, 2000

ADDI R2, R0, 40

ADDI R4, R0, 1000

LOOP: LW R3, 0(R1)

SUBI R2, R2, 4

ANDI R9, R3, 7

BNE R9, R0, LABEL0

ANDI R10, R3, 3

ADDI R6, R10, 12

JUMP LABEL1

SW R10, 0(R1)

LABEL0: JUMP LABEL2

SW R20, 0(R1)

LABEL1: ADDI R1, R1, 4

LABEL 2: BNE R2, R0, LOOP

ANDI R10, R10, 7

SW R10, 4(R2)

**2.ב**

BF=25%, TAKEN=80%

|  |  |  |
| --- | --- | --- |
| מחשב א' | מחשב ב' | מחשב ג' |
| שיטה:PT  BRANCH RESOLUTION: ID  Penalty=|2-1|=1 | מעבד:SPARC  BRANCH RESOLUTION: MEM  Penalty=|4-1|=3 | שיטה:BTB  BRANCH RESOLUTION: EXE  Penalty=|3-1|=2 |

1. **CPI מחשב א'-**

CPI-pnt=CPI-ideal+BF\*penalty\*(n\_taken\*taken\*z)=1+0.25\*1\*(0.2+0.8\*z)=1.05+0.2z

כאשר z מייצג את אי ההצלחה של הקומפיילר במילוי delay slots.

**CPI מחשב ב'-**

-נתון- CB=60%

-נתון כי הקומפיילר מצליח למלא 80% מהdelay slots, ולכן z=0.2.

BFnew=BF\*CB=0.25\*0.60=0.15-

CPI-sarc\_pt=CPI-ideal+BFnew\*penalty\*(n\_taken+taken\*z)=1+0.15\*3\*(0.2+0.8\*0.2)=1.162-

נדרוש שCPI של מחשב א' יהיה קטן מזה של מחשב ב':

כלומר, אחוז האי הצלחה במילוי הקומפיילר חייב להיות קטן מ56% ולכן אחוז ההצלחה חייב להיות לפחות 44%.

1. **CPI מחשב ב'-**

-נתון- CB=60%

-נתון כי הקומפיילר מצליח למלא 80% מהdelay slots, ולכן z=0.2.

BFnew=BF\*CB=0.25\*0.60=0.15-

CPI-sarc\_pt=CPI-ideal+BFnew\*penalty\*(n\_taken+taken\*z)=1+0.15\*3\*(0.2+0.8\*0.2)=1.162

**CPI מחשב ג'-**

CPI-btb=CPI-ideal+BF\*penalty\*BMP=1+0.25\*2\*BMP

נדרוש שCPI של מחשב ב' יהיה קטן מזה של מחשב ג':

לכן BMP חייב להיות לפחות 32.4%.

1. **CPI מחשב א'-**

-נתון כי הקומפיילר מצליח למלא 90% מהdelay slots, ולכן z=0.1.

CPI-pnt=CPI-ideal+BF\*penalty\*(n\_taken\*taken\*z)=1+0.25\*1\*(0.2+0.8\*z)=1.05+0.2\*0.1=1.07

**CPI מחשב ב'-**

-נתון כי הקומפיילר מצליח למלא 80% מהdelay slots, ולכן z=0.2.

BFnew=BF\*CB=0.25\*CB=0.25CB-

CPI-sarc\_pt=CPI-ideal+BFnew\*penalty\*(n\_taken+taken\*z)=1+0.25CB\*3\*(0.2+0.8\*0.2)=1+0.27CB

נדרוש שCPI של מחשב ב' יהיה קטן מזה של מחשב א':

לכן הCB חייב להיות במקסימום 25.9%.

1. CPI מחשב א':

נתון כי הקומפיילר מצליח למלא 90% מהdelay slots, ולכן z=0.1.

CPI-pnt=CPI-ideal+BF\*penalty\*(n\_taken\*taken\*z)=1+0.25\*1\*(0.2+0.8\*z)=1.05+0.2\*0.1=1.07

CPI מחשב ג'-

CPI-btb=CPI-ideal+BF\*penalty\*BMP=1+0.25\*2\*BMP

נדרוש שCPI של מחשב ג' יהיה קטן מזה של מחשב א':

לכן הBMP חייב להיות במקסימום 14%.

**3.א**

0)ADDI R2, R0, 10

4)ADDI R3, R0, 0

8)LOOP: LW R10, 1000(R3) R10= MEM [1000+R3] =60

12) LW R20, 2000(R3) R20= MEM [2000+R3] =50

16) SLT R1, R10, R20 if(R10<R20)-> R1=1 and go to Label0

20) BNE R1, R0, LABEL0

24) SUB R10, R10, R20 R10-=R20=10

28) JUMP LABEL1

32)LABEL0: SUB R10, R20, R10 R10=R20 -R10 =

36)LABEL1: SW R10, 3000(R3) R10= MEM [3000+R3]=70

40) ADDI R3, R3, 4 R3+=4=8

44) SUBI R2, R2, 1 R2-=1=8

48) BNE R2, R0, LOOP 10 ITERATIONS

|  |  |  |
| --- | --- | --- |
| **Branch number** | Target address | Predication |
| 20 | 32 | 10 |
| 48 | 8 | 10 |

**3.ב BNE R1, R0, LABEL0**

|  |  |  |
| --- | --- | --- |
| מספר פנייה | Predication | H/M |
| 1 | 10 | H |
| 2 | 11 | M |
| 3 | 10 | M |
| 4 | 00 | M |
| 5 | 01 | H |
| 6 | 11 | H |
| 7 | 11 | H |
| 8 | 11 | M |
| 9 | 10 | M |
| 10 | 00 | H |

**BNE R2, R0, LOOP**

|  |  |  |
| --- | --- | --- |
| מספר פנייה | Predication | H/M |
| 1 | 10 | H |
| 2 | 11 | H |
| 3 | 11 | H |
| 4 | 11 | H |
| 5 | 11 | H |
| 6 | 11 | H |
| 7 | 11 | H |
| 8 | 11 | H |
| 9 | 11 | H |
| 10 | 11 | M |

**3.ג**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 20 | 0 | 20 | 5 | 20 | 20 | 15 | 40 | 10 | 10 |

**4.א**

Block size=2^20/8192=128byte=2^7

Log(128)=7

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| כתובת | מס' בלוק | SET | H/M | WAY/ סגמנט | הערות |
| 780,950R | 6101 | 2005 | M | 1 | ירמס בלוק 10197 בסגמנט 1.  1->0->3->2 |
| 1,575,100R | 12305 | 17 | M | 1 | ירמס בלוק 8209 .  1->0->3->2 |
| 263,000R | 2054 | 6 | H | 0 | 0->3->2->1 |
| 1,305,300R | 10197 | 2005 | M | 2 | ירמס בלוק 12,245  2->1->0->3 |
| 1,573,700W | 12294 | 6 | M |  | כתיבה רק לbuffer |
| 1,050,800W | 8209 | 17 | M |  | כתיבה רק לbuffer |
| 1,567,400R | 12245 | 2005 | M | 3 | נרמס 2005  3->2->1->0 |
| 1,049,400R | 8198 | 6 | H | 1 | 1->0->3->2 |
| 2200R | 17 | 17 | H | 3 | 3->1->0->2 |
| 781,000W | 6101 | 2005 | H | 1 | 1->3->2->0 כתיבה לbuffer ולקאש |

**4.ב 1**

לבלוקים 0-2047 יש tag=2047/2048=0 ובלוקים 2048-4095 יש tag=4095/2048=1

ניתן להבין כי לכל 2048 בלוקים בזיכרון יש את אותו הtag כי תוצאת החלוקה ב2048 היא זהה (לאחר עיגול התוצאה).

**4.ב.2**

אורך כתובת היא 37 סיביות וכל מרחב הכתובות מנוצל, כתוצאה מכך גודל הזכרון הראשי הינו . מספר הבלוקים בזכרון הראשי הינו:

ישנם 2048 אינדקסים, לכן מספר הבלוקים שיש להם אינדקס זהה הוא:

**4.ב.3**  רק מציין כי בסוף הקודם נאמר כי גודל הזיכרון הראשי הינו

**5.א**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Imm | Rt | Rs | OP | Format | פקודה |
| n-1 | R1 | R2 | ספציפי | I | MODI |

בפעולה הנ"ל מעניינות אותנו ה סיביות הימניות של R2.

כל המספרים שהם חזקה של 2 הן מהצורה שיש 1 ומימנו אפסים.

אם נבצע n-1 כל הסיביות יתהפכו. נקבל ש-n-1 הן סיביות שכולן 1.

**5.ב**

מכיוון שמשתמשים בopCode!=0 נכניס את n-1 ל-ALU ל-ANDI ביחד עם R2, נקבל את הספרות שמעניינות אותנו מ-R2.

בגלל אופן שפעולת AND מתבצעת תוצאת ה-ALU תהיה R2modn.