|  |  |  |
| --- | --- | --- |
| **BAR-ILAN UNIVERSITY (RA)**  Faculty of Engineering  Ramat-Gan 52900, Israel | **Tel: 03-5317722**  **engbi@mail.biu.ac.il** | אוניברסיטת בר-אילן (ע"ר) הפקולטה להנדסה  רמת-גן 52900 |

מבנה מחשבים ספרתיים

# תשע"ט סמסטר ב' מועד ב'

**83-301**

**מרצה:** פרופ' שמואל וימר

**מתרגל:** מר בנימין פרנקל

* **יש לקרוא היטב את ההוראות.**
* **חובה לענות על כל השאלות.**
* **ציון מקסימלי בבחינה: 100 נקודות.**
* **יש לנמק את כל תשובותיכם**.
* יש להקפיד על כתב יד קריא!
* יש לרשום תשובות בתוך הטבלאות המצורפות במקום שנדרש.
* **חומר עזר מותר בשימוש**: מחשבון, ספר הקורס ושקפי ההרצאות בלבד! כל חומר עזר אחר אסור בהחלט!
* **משך הבחינה:** שלוש שעות.
* **יש לצרף את שאלוני הבחינה למחברת!**

**בהצלחה!**

|  |  |  |
| --- | --- | --- |
| R2, 0(R1) | LD | Loop: |
| R4, 0(R3) | LD |  |
| R2, R2, R4 | DADD |  |
| R4, R4, R2 | DSUB |  |
| R2, 0(R1) | SD |  |
| R4, 0(R3) | SD |  |
| R1, R1, #8 | DADDIU |  |
| R3, R3, #8 | DADDIU |  |
| R2, R5, Loop | BNE |  |

**שאלה מספר 1** (55 נקודות):

נתונה התכנית הבאה:

התכנית עוברת על איבריהם של שני מערכים המכילים מספרי integer ומבצעת פעולות אריתמטיות על כל צמד איברים.

נתון מעבד המסוגל להוציא לדרך 2 פקודות במקביל (two-issue processor) וכן תומך ב- dynamic scheduling. הנח כי המעבד מסוגל גם לסיים שתי פקודות מכל סוג שהוא במחזור שעון אחד. במעבד קיימות יחידות ביצוע נפרדות עבור חישוב הכתובת (יחידה אחת), ביצוע פעולות ALU (יחידה אחת) ובדיקת תנאי הקפיצה המותנית (יחידה אחת). נתון כי ביצוע של פעולת ALU אורך מחזור שעון אחד. כמו-כן, נתון כי הזיכרון הינו dual-ported.

* מריצים את התכנית על המעבד הנ"ל כאשר המעבד לא תומך ב- speculation (זאת-אמרת שעובדים על-פי אלגוריתם Tomasulo, אך ללא ROB).

1. מלא את הטבלה הבאה כך שתתאר את התקדמות הפקודות לפי מספר מחזור השעון במהלך ריצת התכנית:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Comment | Write CDB | Mem access | Execute | Issue | Instruction | Loop iteration |
| First issue | 4 | 3 | 2 | 1 | LD R2, 0(R1) | 1 |
| Wait for HW unit | 5 | 4 | 3 | 1 | LD R4, 0(R3) | 1 |
| Wait for LW | 7 |  | 6 | 2 | DADD R2, R2, R4 | 1 |
| Wait for DADD | 9 |  | 8 | 2 | DSUB R4, R4, R2 | 1 |
| Wait for DADD |  | 8 | 4 | 3 | SD R2, 0(R1) | 1 |
| Wait for DSUB |  | 10 | 5 | 3 | SD R4, 0(R3) | 1 |
| Execute directly | 6 |  | 5 | 4 | DADDIU R1, R1, #8 | 1 |
| Wait for ALU | 8 |  | 7 | 4 | DADDIU R3, R3, #8 | 1 |
| Wait for DADD |  |  | 8 | 5 | BNE R2, R5, Loop | 1 |
| Wait for BNE | 11 | 10 | 9 | 6 | LD R2, 0(R1) | 2 |
| Wait for HW unit | 12 | 11 | 10 | 6 | LD R4, 0(R3) | 2 |
| Wait for LW | 14 |  | 13 | 7 | DADD R2, R2, R4 | 2 |
| Wait for DADD | 16 |  | 15 | 7 | DSUB R4, R4, R2 | 2 |
| Wait for DADD |  | 15 | 11 | 8 | SD R2, 0(R1) | 2 |
| Wait for DSUB |  | 17 | 12 | 8 | SD R4, 0(R3) | 2 |
| Execute directly | 11 |  | 10 | 9 | DADDIU R1, R1, #8 | 2 |
| Wait for ALU | 12 |  | 11 | 9 | DADDIU R3, R3, #8 | 2 |
| Wait for DADD |  |  | 15 | 10 | BNE R2, R5, Loop | 2 |
| Wait for BNE | 18 | 17 | 16 | 11 | LD R2, 0(R1) | 3 |
| Wait for HW unit | 19 | 18 | 17 | 11 | LD R4, 0(R3) | 3 |
| Wait for LW | 21 |  | 20 | 12 | DADD R2, R2, R4 | 3 |
| Wait for DADD | 23 |  | 22 | 12 | DSUB R4, R4, R2 | 3 |
| Wait for DADD |  | 22 | 18 | 13 | SD R2, 0(R1) | 3 |
| Wait for DSUB |  | 24 | 19 | 13 | SD R4, 0(R3) | 3 |
| Wait for BNE | 17 |  | 16 | 14 | DADDIU R1, R1, #8 | 3 |
| Wait for ALU | 18 |  | 17 | 14 | DADDIU R3, R3, #8 | 3 |
| Wait for DADD |  |  | 22 | 15 | BNE R2, R5, Loop | 3 |

1. לאיזה ערך שואף ה- CPI עבור מספר גדול של איטרציות?

בין כל איטרציה לאיטרציה הבאה חולפים 7 מחזורי שעון בהם מתבצעות 9 פקודות, ולכן, עבור מספר גדול של איטרציות ה- CPI שואף ל- 

* מהנדס א', מהנדס צעיר בצוות ארכיטקטורה, נדרש לנסות ולשפר את ביצועי המעבד עבור התכנית הנתונה. כמובן, למהנדס ניתנה האפשרות להוסיף חומרה למערכת, ובלבד שהדבר יביא לשיפור בביצועים.

המהנדס החליט לכפיל את מספר יחידות הביצוע, כך שכעת יש למעבד 2 יחידות עבור חישוב הכתובת, 2 יחידות ALU ו- 2 יחידות לבדיקת תנאי הקפיצה המותנית. יחידת הזיכרון נותרה ללא שינוי (dual-ported).

1. מלא את הטבלה הבאה עבור התכנית שרצה על המעבד על-פי הצעתו של מהנדס א':

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Comment | Write CDB | Mem access | Execute | Issue | Instruction | Loop iteration |
| First issue | 4 | 3 | 2 | 1 | LD R2, 0(R1) | 1 |
| First issue | 4 | 3 | 2 | 1 | LD R4, 0(R3) | 1 |
| Wait for LW | 6 |  | 5 | 2 | DADD R2, R2, R4 | 1 |
| Wait for DADD | 8 |  | 7 | 2 | DSUB R4, R4, R2 | 1 |
| Wait for DADD |  | 7 | 4 | 3 | SD R2, 0(R1) | 1 |
| Wait for DSUB |  | 9 | 4 | 3 | SD R4, 0(R3) | 1 |
| Execute directly | 6 |  | 5 | 4 | DADDIU R1, R1, #8 | 1 |
| Wait for ALU | 7 |  | 6 | 4 | DADDIU R3, R3, #8 | 1 |
| Wait for DADD |  |  | 7 | 5 | BNE R2, R5, Loop | 1 |
| Wait for BNE | 10 | 9 | 8 | 6 | LD R2, 0(R1) | 2 |
| Wait for Mem port | 11 | 10 | 8 | 6 | LD R4, 0(R3) | 2 |
| Wait for LW | 13 |  | 12 | 7 | DADD R2, R2, R4 | 2 |
| Wait for DADD | 15 |  | 14 | 7 | DSUB R4, R4, R2 | 2 |
| Wait for DADD |  | 14 | 9 | 8 | SD R2, 0(R1) | 2 |
| Wait for DSUB |  | 16 | 9 | 8 | SD R4, 0(R3) | 2 |
| Execute directly | 11 |  | 10 | 9 | DADDIU R1, R1, #8 | 2 |
| Wait for CDB | 12 |  | 10 | 9 | DADDIU R3, R3, #8 | 2 |
| Wait for DADD |  |  | 14 | 10 | BNE R2, R5, Loop | 2 |
| Wait for BNE | 17 | 16 | 15 | 11 | LD R2, 0(R1) | 3 |
| Wait for Mem port | 18 | 17 | 15 | 11 | LD R4, 0(R3) | 3 |
| Wait for LW | 20 |  | 19 | 12 | DADD R2, R2, R4 | 3 |
| Wait for DADD | 22 |  | 21 | 12 | DSUB R4, R4, R2 | 3 |
| Wait for DADD |  | 21 | 16 | 13 | SD R2, 0(R1) | 3 |
| Wait for DSUB |  | 23 | 16 | 13 | SD R4, 0(R3) | 3 |
| Execute directly | 16 |  | 15 | 14 | DADDIU R1, R1, #8 | 3 |
| Execute directly | 16 |  | 15 | 14 | DADDIU R3, R3, #8 | 3 |
| Wait for DADD |  |  | 21 | 15 | BNE R2, R5, Loop | 3 |

1. לאיזה ערך שואף ה- CPI עבור מספר גדול של איטרציות? האם תוספת החומרה שהוסיף מהנדס א' שיפרה את הביצועים? אם כן, בכמה? אם לא, מדוע?

בין כל איטרציה לאיטרציה הבאה חולפים 7 מחזורי שעון בהם מתבצעות 9 פקודות, ולכן, עבור מספר גדול של איטרציות ה- CPI שואף ל-  . אין שיפור ביצועים למרות שנוספו יחידות ביצוע, וזאת משום שכל איטרציה עדיין לוקחת 7 מחזורי שעון (יש עיכוב בגלל ה- mem access, בגלל תלויות מידע ובגלל פקודת ה- BNE).

* מהנדס ב', מאותו צוות ארכיטקטורה של מהנדס א', ניסה לשפר את ביצועי המעבד בדרך אחרת, וזאת על ידי הרחבת החומרה כך שתוכל להוציא לדרך 3 פקודות במקביל (three-issue processor). הנח כי כעת המעבד מסוגל לסיים שלוש פקודות מכל סוג שהוא במחזור שעון אחד, וכן הנח כי הזיכרון כעת הינו triple-ported. שים-לב, כי על-פי הצעתו של מהנדס ב' ישנה רק יחידת ביצוע אחת מכל סוג, כמו במעבד המקורי המתואר בתחילת השאלה.

1. מלא את הטבלה הבאה עבור התכנית שרצה על המעבד בהתאם להצעתו של מהנדס ב':

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Comment | Write CDB | Mem access | Execute | Issue | Instruction | Loop iteration |
| First issue | 4 | 3 | 2 | 1 | LD R2, 0(R1) | 1 |
| Wait for HW unit | 5 | 4 | 3 | 1 | LD R4, 0(R3) | 1 |
| Wait for LW | 7 |  | 6 | 1 | DADD R2, R2, R4 | 1 |
| Wait for DADD | 9 |  | 8 | 2 | DSUB R4, R4, R2 | 1 |
| Wait for DADD |  | 8 | 4 | 2 | SD R2, 0(R1) | 1 |
| Wait for DSUB |  | 10 | 5 | 2 | SD R4, 0(R3) | 1 |
| Execute directly | 5 |  | 4 | 3 | DADDIU R1, R1, #8 | 1 |
| Wait for ALU | 6 |  | 5 | 3 | DADDIU R3, R3, #8 | 1 |
| Wait for DADD |  |  | 8 | 3 | BNE R2, R5, Loop | 1 |
| Wait for BNE | 11 | 10 | 9 | 4 | LD R2, 0(R1) | 2 |
| Wait for HW unit | 12 | 11 | 10 | 4 | LD R4, 0(R3) | 2 |
| Wait for LW | 14 |  | 13 | 4 | DADD R2, R2, R4 | 2 |
| Wait for DADD | 16 |  | 15 | 5 | DSUB R4, R4, R2 | 2 |
| Wait for DADD |  | 15 | 11 | 5 | SD R2, 0(R1) | 2 |
| Wait for DSUB |  | 17 | 12 | 5 | SD R4, 0(R3) | 2 |
| Wait for BNE | 10 |  | 9 | 6 | DADDIU R1, R1, #8 | 2 |
| Wait for ALU | 11 |  | 10 | 6 | DADDIU R3, R3, #8 | 2 |
| Wait for DADD |  |  | 15 | 6 | BNE R2, R5, Loop | 2 |
| Wait for BNE | 18 | 17 | 16 | 7 | LD R2, 0(R1) | 3 |
| Wait for HW unit | 19 | 18 | 17 | 7 | LD R4, 0(R3) | 3 |
| Wait for LW | 21 |  | 20 | 7 | DADD R2, R2, R4 | 3 |
| Wait for DADD | 23 |  | 22 | 8 | DSUB R4, R4, R2 | 3 |
| Wait for DADD |  | 22 | 18 | 8 | SD R2, 0(R1) | 3 |
| Wait for DSUB |  | 24 | 19 | 8 | SD R4, 0(R3) | 3 |
| Wait for BNE | 17 |  | 16 | 9 | DADDIU R1, R1, #8 | 3 |
| Wait for ALU | 18 |  | 17 | 9 | DADDIU R3, R3, #8 | 3 |
| Wait for DADD |  |  | 22 | 9 | BNE R2, R5, Loop | 3 |

1. לאיזה ערך שואף ה- CPI עבור מספר גדול של איטרציות? האם תוספת החומרה שהוסיף מהנדס ב' שיפרה את הביצועים? אם כן, בכמה? אם לא, מדוע?

בין כל איטרציה לאיטרציה הבאה חולפים 7 מחזורי שעון בהם מתבצעות 9 פקודות, ולכן, עבור מספר גדול של איטרציות ה- CPI שואף ל-  . אין שיפור ביצועים למרות שנוספה חומרה, וזאת משום שכל איטרציה עדיין לוקחת 7 מחזורי שעון (עדיין יש עיכוב בגלל תלויות מידע ובגלל פקודת ה- BNE).

* מהנדס ג', בוגר מצטיין של הפקולטה להנדסה באוניברסיטת בר-אילן, הציע למהנדס ב' להוסיף למעבד תמיכה ב- speculation (עם ROB), וזאת **בנוסף** לשיפורי החומרה שמהנדס ב' כבר הוסיף (three-issue processor, triple-ported memory). שים-לב, כי על-פי הצעה זו ישנה רק יחידת ביצוע אחת מכל סוג, כמו במעבד המקורי המתואר בתחילת השאלה.

1. מלא את הטבלה הבאה עבור התכנית שרצה על המעבד בהתאם להצעתו של מהנדס ג':

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Comment | Commit | Write CDB | Read access | Execute | Issue | Instruction | Loop iteration |
| First issue | 5 | 4 | 3 | 2 | 1 | LD R2, 0(R1) | 1 |
| Wait for HW unit | 6 | 5 | 4 | 3 | 1 | LD R4, 0(R3) | 1 |
| Wait for LW | 8 | 7 |  | 6 | 1 | DADD R2, R2, R4 | 1 |
| Wait for DADD | 10 | 9 |  | 8 | 2 | DSUB R4, R4, R2 | 1 |
| Commit in order | 10 |  |  | 4 | 2 | SD R2, 0(R1) | 1 |
| Commit in order | 10 |  |  | 5 | 2 | SD R4, 0(R3) | 1 |
| Commit in order | 11 | 5 |  | 4 | 3 | DADDIU R1, R1, #8 | 1 |
| Wait for ALU | 11 | 6 |  | 5 | 3 | DADDIU R3, R3, #8 | 1 |
| Wait for DADD | 11 |  |  | 8 | 3 | BNE R2, R5, Loop | 1 |
| Wait for HW unit | 12 | 8 | 7 | 6 | 4 | LD R2, 0(R1) | 2 |
| Wait for HW unit | 12 | 9 | 8 | 7 | 4 | LD R4, 0(R3) | 2 |
| Wait for LW | 12 | 11 |  | 10 | 4 | DADD R2, R2, R4 | 2 |
| Wait for DADD | 14 | 13 |  | 12 | 5 | DSUB R4, R4, R2 | 2 |
| Wait for HW unit | 14 |  |  | 8 | 5 | SD R2, 0(R1) | 2 |
| Wait for HW unit | 14 |  |  | 9 | 5 | SD R4, 0(R3) | 2 |
| Commit in order | 15 | 8 |  | 7 | 6 | DADDIU R1, R1, #8 | 2 |
| Wait for ALU | 15 | 10 |  | 9 | 6 | DADDIU R3, R3, #8 | 2 |
| Wait for DADD | 15 |  |  | 12 | 6 | BNE R2, R5, Loop | 2 |
| Wait for HW unit | 16 | 12 | 11 | 10 | 7 | LD R2, 0(R1) | 3 |
| Wait for HW unit | 16 | 13 | 12 | 11 | 7 | LD R4, 0(R3) | 3 |
| Wait for LW | 16 | 15 |  | 14 | 7 | DADD R2, R2, R4 | 3 |
| Wait for DADD | 18 | 17 |  | 16 | 8 | DSUB R4, R4, R2 | 3 |
| Wait for HW unit | 18 |  |  | 12 | 8 | SD R2, 0(R1) | 3 |
| Wait for HW unit | 18 |  |  | 13 | 8 | SD R4, 0(R3) | 3 |
| Wait for ALU | 19 | 12 |  | 11 | 9 | DADDIU R1, R1, #8 | 3 |
| Wait for ALU | 19 | 14 |  | 13 | 9 | DADDIU R3, R3, #8 | 3 |
| Wait for DADD | 19 |  |  | 16 | 9 | BNE R2, R5, Loop | 3 |

1. לאיזה ערך שואף ה- CPI עבור מספר גדול של איטרציות? האם השינוי שהציע מהנדס ג' תרם לשיפור הביצועים של המעבד? אם כן, בכמה? אם לא, מדוע?

בין כל איטרציה לאיטרציה הבאה חולפים 4 מחזורי שעון בהם מתבצעות 9 פקודות, ולכן, עבור מספר גדול של איטרציות ה- CPI שואף ל-  . יש שיפור ביצועים, וזאת משום שכבר אין עיכוב בגלל פקודת ה- BNE).

**שאלה מספר 2** (55 נקודות):

בשאלה זו נעסוק בתכנון וניתוח ביצועים של מערכות זיכרון.

* נתון מעבד עם מערכת זיכרון בעלת רמה אחת של זיכרון מטמון (L1). נתון כי זמן הגישה הממוצע לזיכרון הוא 2.4 מחזורי שעון, כאשר:
  + אם המידע נמצא בזיכרון המטמון אזי זמן הגישה לזיכרון הוא מחזור שעון יחיד,
  + אם המידע לא נמצא בזיכרון המטמון אזי נדרשים עוד 80 מחזורי שעון על מנת להביא את המידע מהזיכרון הראשי (main memory).

צוות מהנדסים מנסה לשפר את זמן הגישה הממוצע לזיכרון, במטרה להשיג שיפור של 65% בזמן הגישה הממוצע (speedup=1.65). צוות המהנדסים החליט להוסיף רמה נוספת של זיכרון מטמון (L2), כאשר נתון כי:

* + זמן הגישה ל- L2 הינו 6 מחזורי שעון.
  + הוספת ההיררכיה של L2 למערכת הזיכרון איננה משפיעה כלל על רצף הגישות ל- L1 או על זמני הגישות ל- L1.
  + זמן הגישה לזיכרון הראשי נותר כשהיה (80 מחזורי שעון).

1. מה צריך להיות ה- hit rate של L2 על מנת להשיג את השיפור הרצוי ב- AMAT?

ראשית, עלינו למצוא מהו ה- miss rate של L1:



בשלב הבא, נרצה לחשב מהו ה- AMAT שעלינו לחתור אליו על מנת להשיג את השיפור הנדרש בשאלה:

 

כעת, נוכל להשתמש שוב בנוסחת ה- AMAT על מנת למצוא את ה- miss rate של L2 שיניב את השיפור הרצוי:



מפתרון של המשוואה הנ"ל עולה כי ה- miss rate הגבוהה ביותר האפשרי עבור L2 הוא:  , ולכן, ה- hit rate של L2 צריך להיות לפחות: 

* נתונה מערכת בעלת מעבד יחיד (single core system). המעבד הינו מצונר (pipelined), כך שה- CPI האידאלי שלו הינו מחזור שעון יחיד. שים-לב, כי ה- CPI האידאלי איננו כולל את התקורה (overhead) הנובעת מהחטאות בזיכרון המטמון (cache misses).

על המערכת הנ"ל הריצו benchmark מסוים, כאשר מערכת הזיכרון של המעבד הכילה L1 cache. מתוך ניתוח ריצת ה- benchmark עולה כי עבור כל 10,000,000 גישות לזיכרון המטמון היו 308,752 החטאות (L1 cache misses), כאשר:

* + אם המידע נמצא בזיכרון המטמון אזי זמן הגישה לזיכרון הוא מחזור שעון יחיד, ולא נוצר שום עיכוב ב- pipeline.
  + אם המידע לא נמצא בזיכרון המטמון אזי נדרשים עוד 10 מחזורי שעון על מנת להביא את המידע.

ראש צוות הארכיטקטורה בחברה שבה משתמשים במערכת המתוארת לעיל (single core system) קיבל החלטה להחליף את המערכת הנ"ל במערכת מרובת ליבות (multi-core system). במערכת החדשה לכל core יש זיכרון מטמון (L1) משלו, כאשר נתון כי:

* + כל הליבות (cores) מתייחסות לזיכרון מרכזי משותף אחד,
  + התנגשויות פוטנציאליות עבור נתונים משותפים מטופלות בשיטת snooping ובפרוטוקול MSI עבור cache coherence.

על המערכת החדשה הריצו את אותו benchmark שבו השתמשו עבור ניתוח ביצועי המערכת הישנה, ומתוך ניתוח ריצת ה- benchmark עולה כי כעת ישנם 452,977 החטאות (L1 cache misses) עבור כל 10,000,000 גישות לזיכרון המטמון, כאשר:

* + אם המידע נמצא בזיכרון המטמון אזי זמן הגישה לזיכרון נותר מחזור שעון יחיד.
  + אם המידע לא נמצא בזיכרון המטמון אזי נדרשים עוד 14 מחזורי שעון בממוצע על מנת לטפל ב- L1 cache miss שאירע.

1. מה חייב להיות ערכו של ה- CPI האידאלי של ה- multi-core system על מנת שהחלטתו של ראש צוות הארכיטקטורה (לעבור מה- single core system ל- multi-core system) תהיה כדאית מבחינת ביצועים? האם הדבר אפשרי? הסבר.

ראשית, נרצה לחשב את ה- CPI של המערכת בעלת מעבד יחיד. לשם כך, נחשב את התקורה הנובעת מהחטאות בגישה לזיכרון L1, ונקבל כי:



על מנת שההחלטה לעבור ל- multi-core system תהיה כדאית מבחינת ביצועים, עלינו לדאוג שיתקיים:



ולכן, ה- CPI האידאלי צריך להיות קטן מ- 0.6745842. הדרישה הזאת אפשרית בהחלט, שהרי מדובר במערכת מרובת ליבות העובדות במקביל, ולכן אפשרי להשיג CPI שכזה.

* נדרש לתכנן זיכרון מטמון היררכי. על זיכרון המטמון להיות בעל **Inclusive policy**. נתון כי גודלו של כל בלוק בזיכרון הינו 16B. הוחלט כי ה- L1 cache יהיה בגודל של , משום שזהו הגודל **המקסימלי** של היררכיה בזיכרון המטמון שאליה ניתן לגשת במחזור שעון יחיד. לדוגמא, אם היו מוסיפים ל- L1 עוד בלוק אחד נוסף הרי שכבר היו נדרשים שני מחזורי שעון על מנת לגשת ל- L1.

כעת, רוצים להוסיף היררכיית זיכרון מטמון נוספת (L2), כאשר נתון כי:

* + זמן הגישה להיררכיה בזיכרון המטמון הינו לינארי בגודל ההיררכיה, כך שזמן הגישה להיררכיה בגודל  הינומחזורי שעון. כמובן, שאםאיננו מספר שלם, הרי שמספר מחזורי השעון הנדרש לצורך גישה להיררכיה הוא .
  + ה- Miss Rate של ההיררכיה ה-  בזיכרון המטמון נתון ע"י הביטוי:

,

כאשר  הינו מספר הבתים (Bytes) בהיררכיה ה-  , וכן מוגדר כי: 

* + זמן הגישה ל- main memory הינו 100 מחזורי שעון.

1. עליך לקבוע את  כך שיתקבל AMAT מינימלי. לשם כך, אין הגבלה על כמות החומרה. פרט ונמק את צעדיך. רשום את ערכו המדויק של ה- AMAT המינימלי המתקבל עבור  שמצאת.

ראשית, נחשב את ה- Miss rate של כל אחת מההיררכיות, וכן את זמן הגישה לכל היררכיה:









הביטוי הכללי של זמן הגישה הממוצע לזיכרון הוא:



נציב ונקבל משוואה עם נעלם אחד: 

זוהי פונקציה שאינה רציפה, משום שהביטוי של  תלוי בגודל של L2 באופן לא רציף. בנוסף, גם  בעצמו איננו משתנה רציף, שהרי הוא חייב להיות כפולה שלמה של 16 (גודל בלוק).   
בשלב הראשון נפתור את הבעיה הרציפה ונניח כאילו  תלוי בגודל של L2 באופן רציף, וכן כי  הינו משתנה רציף. לאחר מכן, נתחשב בעובדה כי המציאות מכתיבה פונקציה לא רציפה.   
נגזור את הפונקציה הרציפה ונשווה את הנגזרת לאפס על מנת למצוא מינימום:







זוהי נקודת האופטימום של הפונקציה הרציפה. כמובן, שאין כל אפשרות להגדיר היררכיה של זיכרון עם שברי בתים, וגם לא עם שברי בלוקים, ולכן, נמצא את מספר הבלוקים השלם הקרוב לערך הזה. אם נעגל כלפי מטה נקבל:







אם נעגל את מספר הבלוקים כלפי מעלה נקבל תוצאה אפילו קצת יותר טובה:







זה יכול היה להיות ה- AMAT **אם**  היה יכול להיות מספר לא שלם של מחזורי שעון, אך מכיוון שבפועל אנחנו משלמים רק מספר שלם של מחזורי שעון, הרי שעבור  אנחנו נשלם 8 מחזורי שעון.



מובן כי אין כל היגיון לקבוע את , שהרי באותו מחיר (של 8 מחזורי שעון) אנחנו יכולים להוסיף עוד 1383 בלוקים, שהם 22128 בתים, ובכך להקטין את ה- miss rate של L2 וזאת מבלי לשלם שום מחיר מבחינת זמן הגישה ל- L2.

נדרש להבין כי תנאי הבעיה מגדירים פונקציה דיסקרטית (מתוארת בגרף בכחול), אשר ערכה מתלכד עם ערך הפונקציה הרציפה (מתוארת בגרף באדום) במקומות שבהם  הינו כפולה שלמה של :



לכן, כל שנותר לעשות הוא רק לבדוק איזו אפשרות מבין שתי האפשרויות שהן כפולה שלמה של הסמוכות לנקודת המינימום של הפונקציה הרציפה נותן AMAT טוב יותר:





ולכן, נקבע את להיות  , ובכך נשיג: 