**חלק יבש**

**2.1**

מימוש הבורר:

Diagram, engineering drawing

Description automatically generated

טבלת אמת:

|  |  |  |  |
| --- | --- | --- | --- |
| **d0** | **d1** | **sel** | **z** |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

ביטוי לZ כפונקציה של הכניסות :

חישוב השהיות:

נעזר במפת קרנו ע"מ למצוא את הצירופים עבורם המוצא משתנה בשינוי כניסה יחידה:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **d0d1**  **sel** | **00** | **01** | **11** | **10** |
| **0** | 0 | 0 | 1 | 1 |
| **1** | 0 | 1 | 1 | 0 |

נתחיל מחישוב .

הצירופים שגורמים לשינוי במוצא הם:

1. d0 = 0, d1 = 1, sel = 0->1:

    TPDlh(sel->z) = TPDhl(NOT)+TPDhl(OR)+TPDlh(NOT)+TPDlh(OR)=24ns

1. d0 = 0, d1 = 1, sel = 1->0:

    TPDhl(sel->z) = TPDlh(NOT)+TPDlh(OR)+TPDhl(NOT)+TPDhl(OR)=24ns

1. d0 = 1, d1 = 0, sel = 0->1:

TPDhl(sel->z) = TPDlh(OR)+TPDhl(NOT) + TPDhl(OR) = 14ns

1. d0 = 1, d1 = 0, sel = 1->0:

    TPDlh(sel->z) = TPDhl(OR)+TPDlh(NOT)+TPDlh(OR)=15ns

חישוב TPD(d0->z):

1. D0 =0->1, d1 = 0, sel = 0:

    TPDlh(d0->z) = TPDhl(NOT)+TPDhl(OR)+TPDlh(NOT)+TPDlh(OR)=24ns

1. D0=1->0 ,d1 = 0, sel = 0:

    TPDhl(d0->z) = TPDlh(NOT)+TPDlh(OR)+TPDhl(NOT)+TPDhl(OR)=24ns

טבלת זמני ההשהיה שקיבלנו:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 9 | 10 | NOT |
| 4 | 1 | OR2 |
| 4 | 6 | XOR2 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **path** | **d0** | **d1** | **sel** | **TPD** |
| sel->g2->g4->g6->g7 | 0 | 1 | 0->1 | 24 |
| sel->g2->g4->g6->g7 | 0 | 1 | 1->0 | 24 |
| sel->g3->g5->g7 | 1 | 0 | 0->1 | 14 |
| sel->g3->g5->g7 | 1 | 0 | 1->0 | 15 |
| d0->g0->g3->g5->g7 | 0->1 | 0 | 0 | 24 |
| d0->g0->g3->g5->g7 | 1->0 | 0 | 0 | 24 |
| d0->g0->g3->g5->g7 | 0->1 | 1 | 0 | 24 |
| d0->g0->g3->g5->g7 | 1->0 | 1 | 0 | 24 |
| d1->g1->g4->g6->g7 | 0 | 0->1 | 1 | 24 |
| d1->g1->g4->g6->g7 | 0 | 1->0 | 1 | 24 |
| d1->g1->g4->g6->g7 | 1 | 0->1 | 1 | 24 |
| d1->g1->g4->g6->g7 | 1 | 1->0 | 1 | 24 |

**2.2**

מימוש הבורר:

Diagram, schematic

Description automatically generatedנבחר את המצב הבא:

נחשב את זמן ההשהיה עבור .

לפי הטבלה בסעיף קודם:

עבור :

**2.3**

טבלת ההשהיות החדשה:

|  |  |  |
| --- | --- | --- |
| TPDhl | TPDlh |  |
| 10 | 10 | NOT |
| 4 | 4 | OR2 |
| 6 | 6 | XOR2 |

נבנה מפת קרנו עבור כל אחת מהיציאות:

**s**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **a b**  **cin a\_ns** | **00** | **01** | **11** | **10** |
| **00** | 0 | 1 | 0 | 1 |
| **01** | 0 | 1 | 0 | 1 |
| **11** | 1 | 0 | 1 | 0 |
| **10** | 1 | 0 | 1 | 0 |

**cout**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **a b**  **cin a\_ns** | **00** | **01** | **11** | **10** |
| **00** | 0 | 1 | 0 | 0 |
| **01** | 0 | 0 | 1 | 0 |
| **11** | 0 | 1 | 1 | 1 |
| **10** | 1 | 1 | 1 | 0 |

מימוש המעגל:

Diagram, schematic

Description automatically generated

חישוב ההשהיות המקסימליות:

**2.4**

המימוש שלנו ליחידה:

Diagram

Description automatically generated

עבור op=00, נקבל ביציאת s את d0 שזהו בדיוק a NOT b.

עבור op=01, נקבל ביציאת s את d1 שזהו בדיוק a XOR b.

עבור op=10, נקבל בכניסת a\_ns ברכיב ה-FAS את הערך 1 ולכן ביציאה s נקבל את d2 שזהו בדיוק הערך a+b+Cin.

עבור op=11, נקבל בכניסת a\_ns ברכיב ה-FAS את הערך 0 ולכן ביציאה s נקבל את d3 שזהו בדיוק הערך a-b-Cin.

השתמשנו בבורר אחד, ביחידת FAS אחת ובשערי הספריה בלבד.

חישוב השהיות:

נסדר את הזמנים הזמנים בטבלה הבאה.

כל ערך בטבלה הוא זמן ההשהייה המקסימלי מהכניסה ליציאה הרלוונטית.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **op[1]** | **op[0]** | **Cin** | **B** | **A** | Tpd |
| 24 | 48 | 54 | 62 | 62 | **s** |
| 0 | 24 | 32 | 32 | 28 | **Cout** |

הסבר על החישובים:

לפי הטבלה בסעיף 2.2, עבור שינוי של [op[0 ושאר הכניסות קבועות, נקבל במקרה הגרוע .

באופן דומה עבור שינוי של [op[1, נקבל במקרה הגרוע .

לפי החישובים של סעיף 2.3, מתקיים:

עבור Cin, חישובהוא לפי החישוב בסעיף 2.3 שהוא 32.

חישוב הוא לפי סעיף 2.3 (שהוא 6) + ההשהיה המקסימלית של הבורר עבור שינוי בערכי ה-DATA שהיא 48 ובסך הכל 54 יחידות זמן.

עבור b, חישוב הוא לפי סעיף 2.3 - 32 יחידות זמן.

חישוב הוא לפי המעבר בלוגיקה הצירופית OR+NOT כי זהו המסלול עם ההשהיה הארוכה ביותר - 14 יחידות זמן, ונוסף על זה ההשהייה המירבית של הבורר בעת שינוי כניסת DATA - 48 יחידות זמן ובסך הכל 62 יחידות זמן.

עבור a באופן דומה: .

**2.5**

ניתן לממש רכיב זה באמצעות שימוש ב-64 רכיבי ALU כמו בסעיף קודם, באופן הבא:

הרכיב ה- יחובר כך שכניסות ה-a וה-b שלו יחוברו לסיבית המתאימה בוקטורי הכניסה ו-, יציאת ה-s שלו תחובר לסיבית המתאימה בוקטור היציאה - .

כמו כן כניסת ה-c\_in תחובר ליציאת c\_out של רכיב ה-ALU הקודם (מלבד הרכיב הראשון). כניסת ה-op תחובר ישירות לכניסת ה-op של רכיב ה-ALU הגדול.

שרטוט של המימוש:

Diagram

Description automatically generated

הסבר לנכונות המימוש:

עבורת פעולת NOR או XOR, הם מתבצעות באופן bitwise ולכן הפעולה צריכה להתבצע על כל ביט בנפרד והתוצאה צריכה להיכנס לביט המתאים בוקטור התוצאה, זה בדיוק מה שקורה במימוש זה. זמן ההשהייה עבור פעולות אלו הוא T\_pd(ALU) עבור השינוי המתאים, מכיוון שפעולות זו קורות באופן מקבילי.

עבור פעולות חיבור וחיסור, מתחילים מביצוע הפעולה על הביט הימני ביותר (LSB), ולאחר מכן ממשיכים לביצוע הפעולה על הביט הבא, כאשר ה-carry מהפעולה הקודמת נכנס בתור ה-c\_in בפעולה זו. כך הפעולה מתבצעת בצורה נכונה על המספר השלם (הוקטור) ולא bitwise. פעולות אלו מתבצעות בצורה טורית, לכן זמן ההשהייה מחושב בצורה טורית, כי צריך לחכות לתוצאת ה-c\_out של הרכיב הקודם.

אז זמן ההשהיה המקסימלי יתקבל עבור הערכים הבאים:

op = 10

a = 111…1

b = 111…1

c\_in = 0

ונסתכל על הזמן הבא, בעת שינוי b[0]: 0->1:

T\_pd(b[0]->s[63]) = 32 + 32\*62 + 54 = 2070

זאת כיוון שבALU0 אנו מסתכלים על הזמן (T\_pd(b->c\_out ועבור ALU1-62 מסתכלים על הזמן (T\_pd(c\_in->c\_out ולבסוף ב-ALU63 מסתכלים על הזמן (T\_pd(c\_in->s. סכימה של כולם מביאה 2070 יחידות זמן, זהו זמן ההשהייה המקסימלי.