# LAB 3

**מגישים** :

- יגל בן צבי 208584615
- רון בניטה 314882317

**תאריך הגשה :** 11.05.2025

**מחלקה :** המחלקה להנדסת חשמל

**קורס :** מעבדת ארכיטקטורה מתקדמת ומאיצי חומרה - 361.1.4693

אוניברסיטת בן-גוריון בנגב
Ben-Gurion University of the Negev

# הקדמה

במעבדה זו אנו נדרשים לפתח בקר בסיסי מסוג MULTI-CYCLE על מנת להריץ תוכנית. הבקר מחולק ל-2 חלקים עיקרים : control unit (FSM) and datapath כך שה – control unit משמש כ"ראש" וה – datapath משמש כשרירים שמבצעים את הפעולות שהראש מנחה אותם.

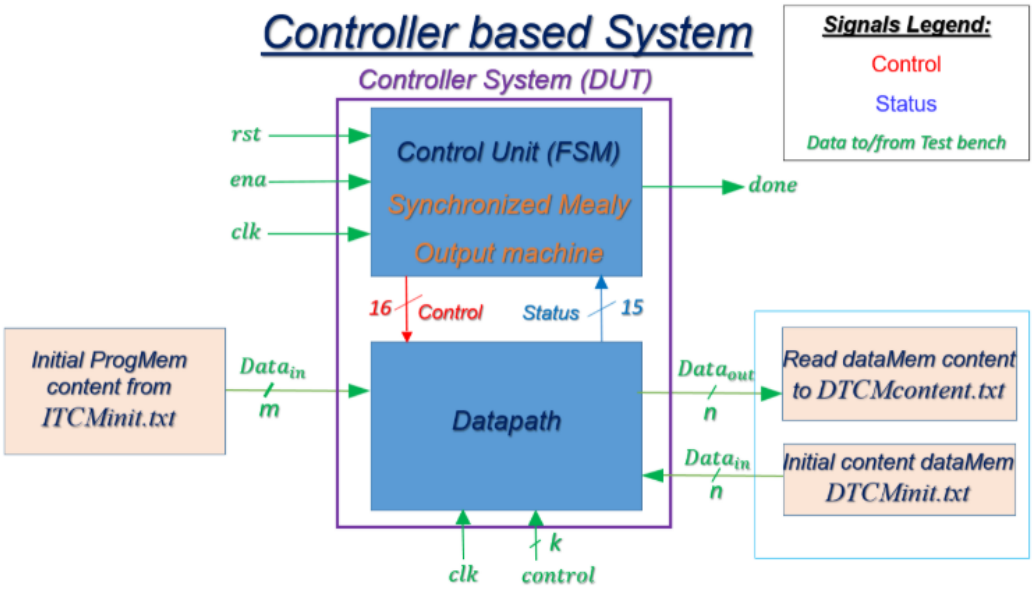



**Figure 1: Overall DUT structure**

הבקר יתמוך בפקודות המפורטות ב – ISA הבאה :

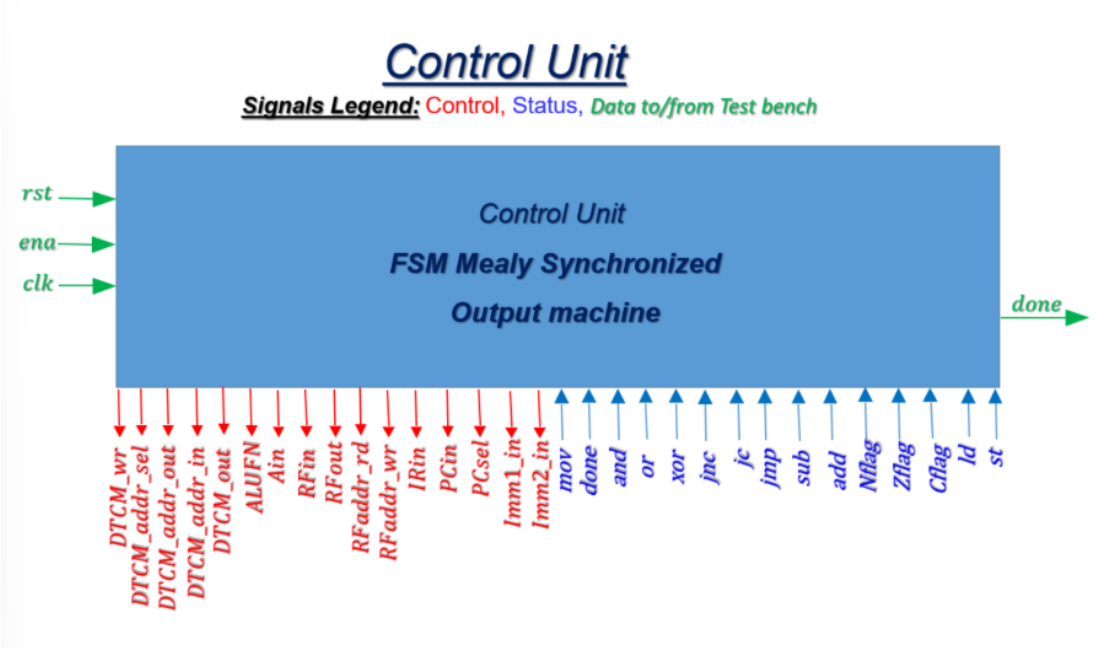
| Instruction Format | Decimal value | OPC | Instruction | Explanation | N | Z | C |
|---|---|---|---|---|---|---|---|
| R-Type | 0 | 0000 | add ra,rb,rc | R[ra]<=R[rb]+R[rc] | * | * | * |
| | | | nop | R[0]<=R[0]+R[0] (*emulated instruction*) | * | * | * |
| | 1 | 0001 | sub ra,rb,rc | R[ra]<=R[rb]-R[rc] | * | * | * |
| | 2 | 0010 | and ra,rb,rc | R[ra]<=R[rb] and R[rc] | * | * | - |
| | 3 | 0011 | or ra,rb,rc | R[ra]<=R[rb] or R[rc] | * | * | - |
| | 4 | 0100 | xor ra,rb,rc | R[ra]<=R[rb] xor R[rc] | * | * | - |
| | 5 | 0101 | *unused* | | | | |
| | 6 | 0110 | *unused* | | | | |
| J-Type | 7 | 0111 | jmp offset_addr | PC<=PC+1+offset_addr | - | - | - |
| | 8 | 1000 | jc /jhs offset_addr | If(Cflag==1) PC<=PC+1+offset_addr | - | - | - |
| | 9 | 1001 | jnc /jlo offset_addr | If(Cflag==0) PC<=PC+1+offset_addr | - | - | - |
| | 10 | 1010 | *unused* | | | | |
| | 11 | 1011 | *unused* | | | | |
| I-Type | 12 | 1100 | mov ra,imm | R[ra]<=imm | - | - | - |
| | 13 | 1101 | ld ra,imm(rb) | R[ra] <= M[imm+R[rb]] | - | - | - |
| | 14 | 1110 | st ra,imm(rb) | M[imm+R[rb]] <= R[ra] | - | - | - |
| Special | 15 | 1111 | done | Signals the TB to read the DTCM content | - | - | - |

Note: * The status flag bit is affected , - The status flag bit is not affected

## Control unit:

הוא משמש כ"מוח" של הבקר, ממוש בעזרת מכונת מצבים סופית (FSM) , הוא מקבל אותות כניסה מ-2 מקורות : הראשון זה מה-TB (בירוק) והשני זה מה-datapath (בכחול).

והוא מוציא אותות וחיווים אל ה- datapath (באדום).

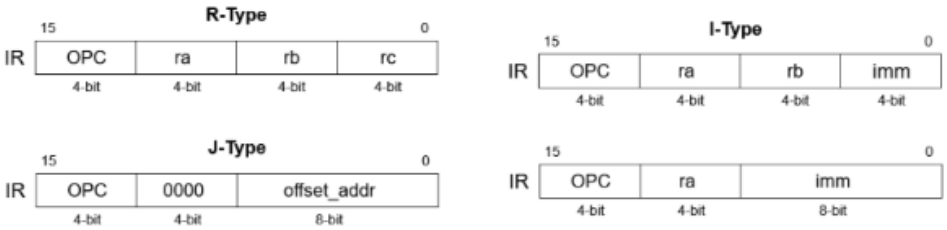



בעזרת האותות היוצאים אל ה – datapath אנו מחליטים כיצד הידע יעבור, יעובד, ומי יכתוב ויקרא מאיפה ולאן.

אנו ממשנו בעזרת 2 מצבים בסיסים לכל פקודה שהם :

- Fetch – הבאה של הפקודה הבאה
- Id (decode) – פענוח הפקודה

ומכאן חילקנו את המצב האחרון שהוא excute כלומר ביצוע הפקודה בהתאם לסוגי הפקודות :




- R-type
- I-type
- J-type

## Data Path:

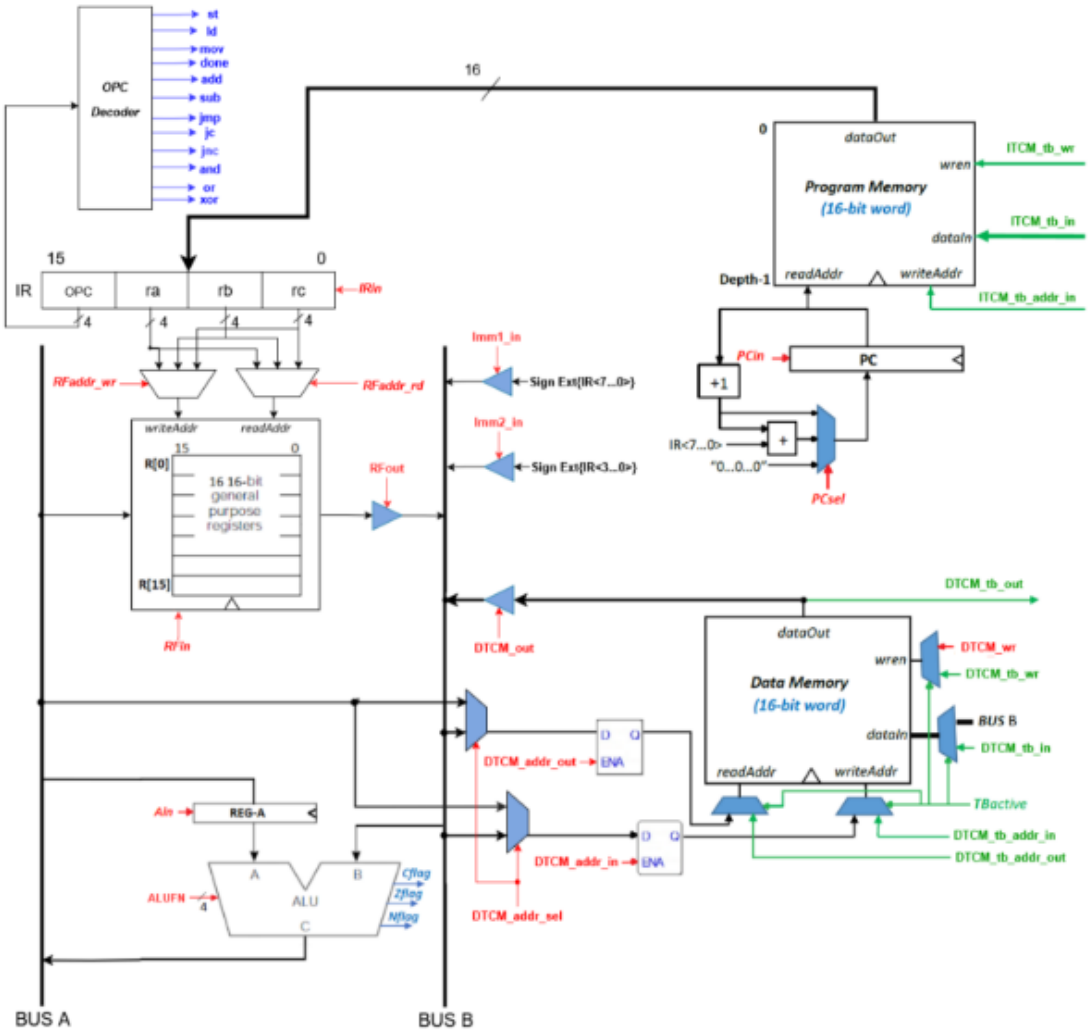
השרירים אשר מבצעים את מה שיחידת השליטה מורה להם לבצע.

בחלק זו אנו מיישמים באופן מקבילי, ומתבססים על הידע שצברנו ב-2 המעבדות הקודמות.

אנו בעצם בונים מערכת אשר מקבלת פקודות מהזיכרון באופן טורי, מביאה אותם (fetch), מפענחת אותם (decode) ומבצעת אותם (execute).

ישנו כמובן זיכרון לפקודות (program memory) וזיכרון למידע ונתונים ( data memory).

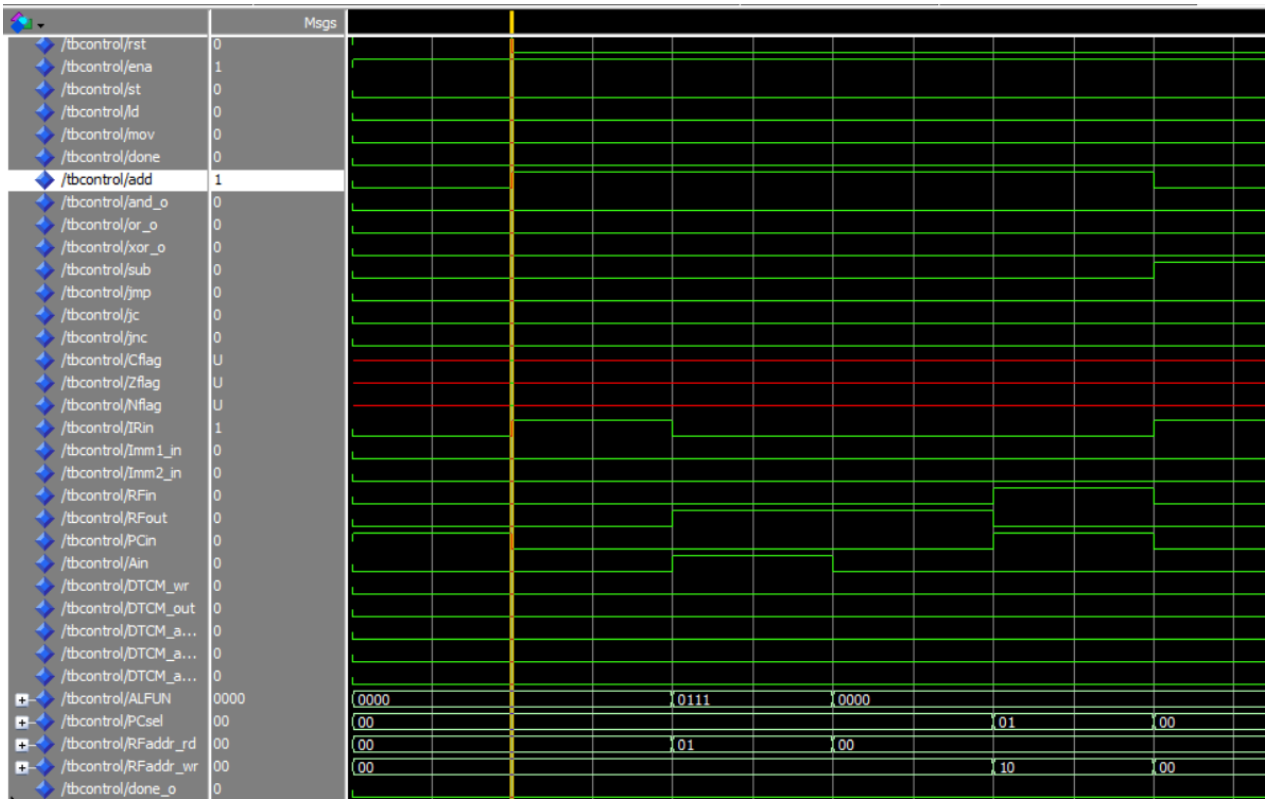
בנוסף אנו עובדים עם 2 קווי תעבורה (bus) : bus A, bus B .



בדיקה של הרכיבים :

בבדיקה זו ניקח דוגמה למימוש מכל סוג פקודה R-type, J-type, I-type

עבור R_type ניקח את פקודות ADD, נדליק את קו הבקרה של ADD ונראה כי אנו אכן עוברים בין המצבים כמצופה.




שלב 1 : fetch - בשלב זה רק הביט של ADD דולק (זו המטרה) וכי Irin דולק גם על מנת להכניס את הפקודה הבאה שהיא על ה-bus לתוך רגיסטר IR.

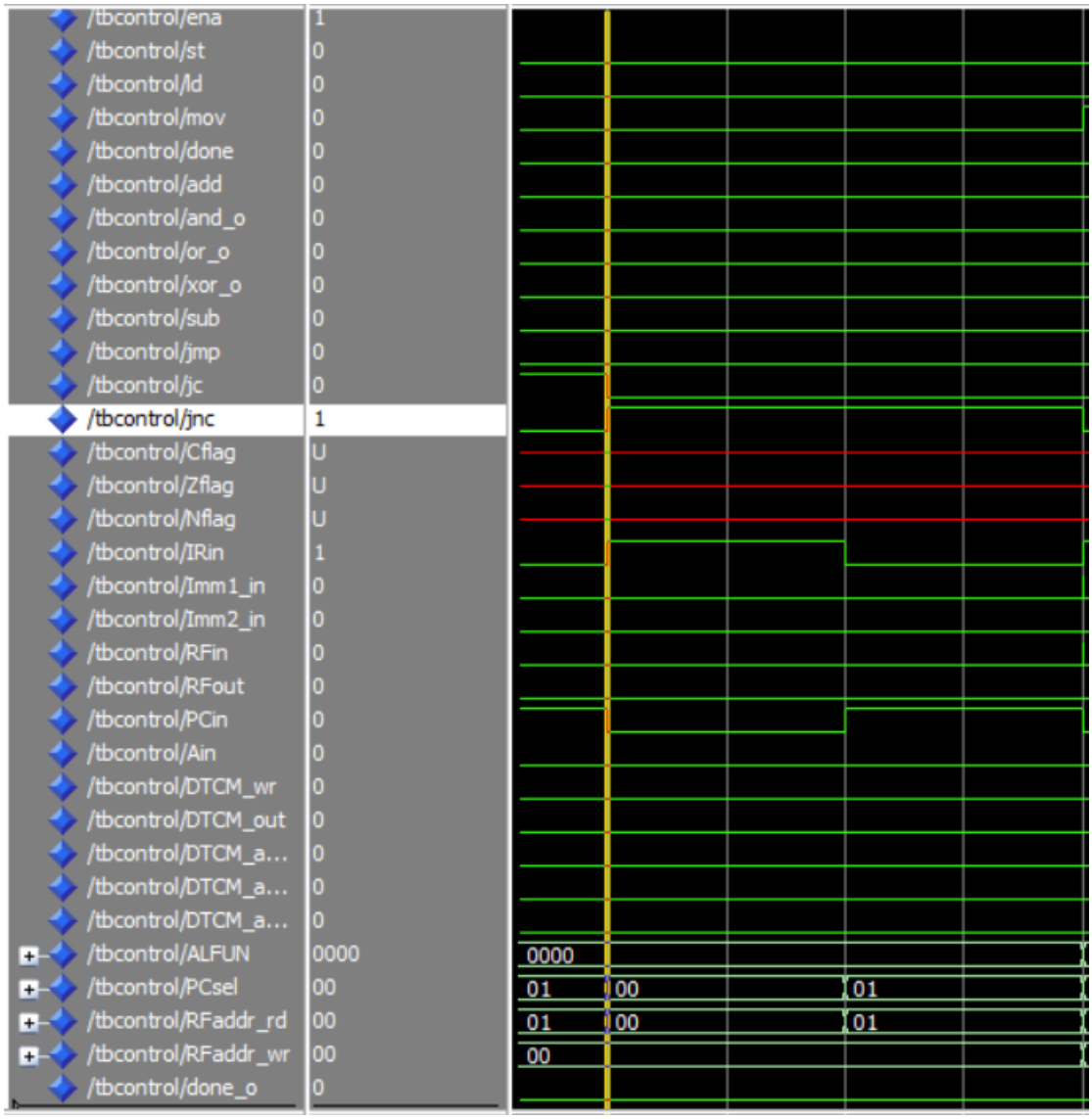
שלב 2 : id(decode) – בשלב זה נדלקים רק ה- RFout,Ain שבו אנו מוצאים את הרגיסטר המתאים על קו ה-A אל רגיסטר A בכניסה ל-ALU.

נדלקים גם ALUFN = 0111 שזה אומר שמה שעל busB יועבר אל c וכי RFaddr_in גם דולק כדי להכניס את הכתובת המתאימה.

שלב 3 :execute שזה שלב הביצוע בו אנו מבצעים את פעולת הADD ב-ALU פה רק RFout דולק. ובמחזור שעון הבאה יידלקו הביטים

PCin, RFin ו-PCsel = 01,RFaddr_wr = 10.

עבור J_type ניקח את פקודות jnc, נדליק את קו הבקרה של jnc ונראה כי אנו אכן עוברים בין המצבים כמצופה.




שלב 1 : fetch - בשלב זה רק הביט של jnc דולק (זו המטרה) וכי Irin דולק גם על מנת להכניס את הפקודה הבאה שהיא על ה-bus לתוך רגיסטר IR.

שלב 2 : id(decode) and execute– בשלב זה נדלקים PCin

ו – PC_sel = 01=RFaddr_rd.

ומכיוון שאין carry אז אנו מסיימים פה ללא קפיצה והכנסה של הקפיצה.