

ADVANCED CPU ARCHITECTURE AND HARDWARE ACCELERATORS LABRATORY

Preparation Report LAB 3

Dvir Zaguri 315602284

Elad Hubashi 313612038

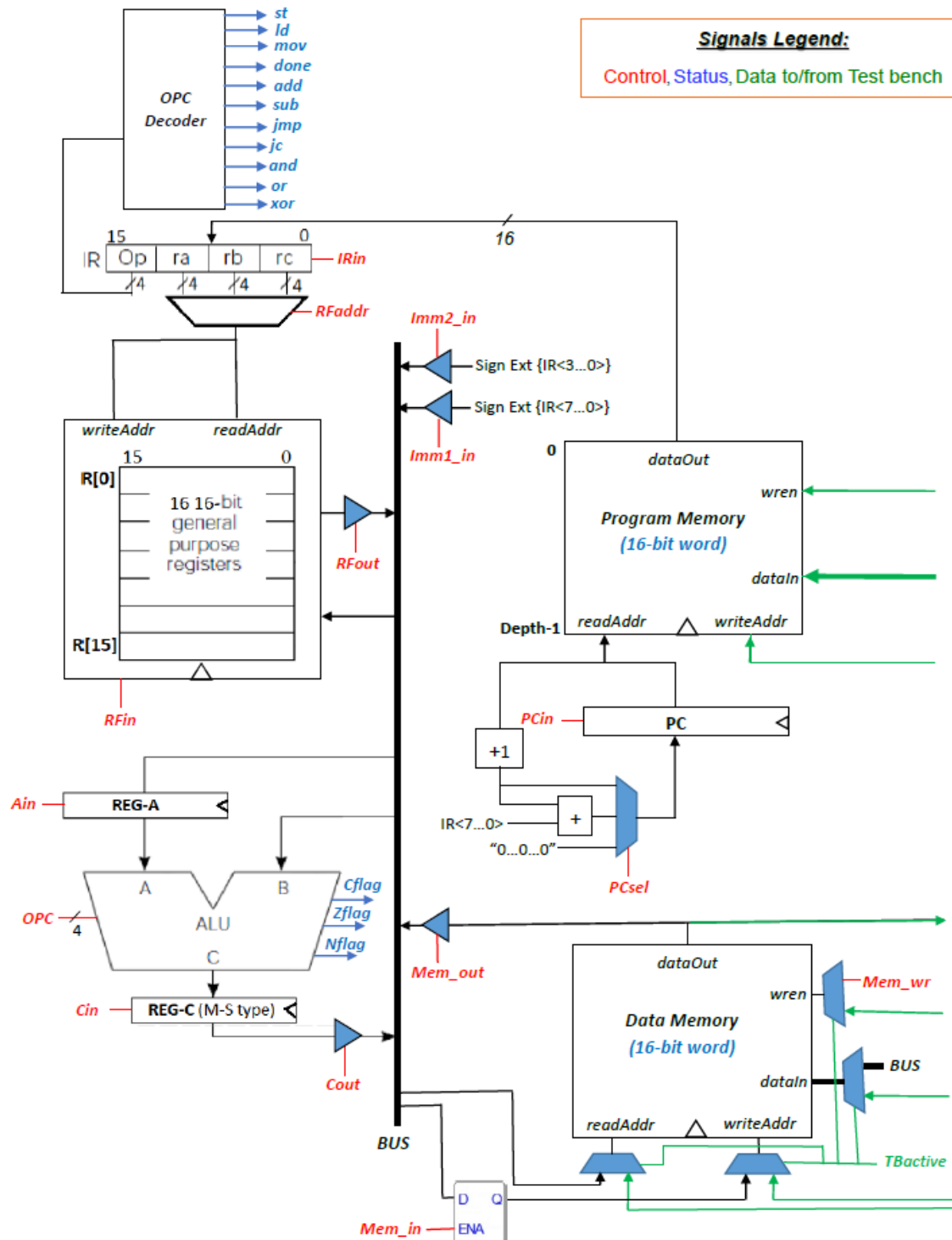
רקע

מטרות המעבדה הן רכישת כישורי כתיבת קוד מקבילי ב-VHDL תוך שימוש ב-Sequential logic, עיצוב מתודולוגיית Control ו-Datapath בנפרד, והכנה למעבדה בה נצטרך לכתוב סינטזה ל-FPGA. במטלת הדו"ח המכין הזה נממש מערכת המורכבת משכבת מעטפת (top) המכילה רכיבים רבים. המערכת תעבוד בצורה סינכרונית ותחולק למספר חלקים כאשר כל אחד מהם יהיה מקבילי ויוסבר בהמשך. כל תהליך במערכת יבצע פעולה לוגית / חישובית שונה בכדי יחד לזהות רצף ביטים מצורה מסויימת לפי החוקיות מטה.

פירוט ה-Modules הקיימים במערכת

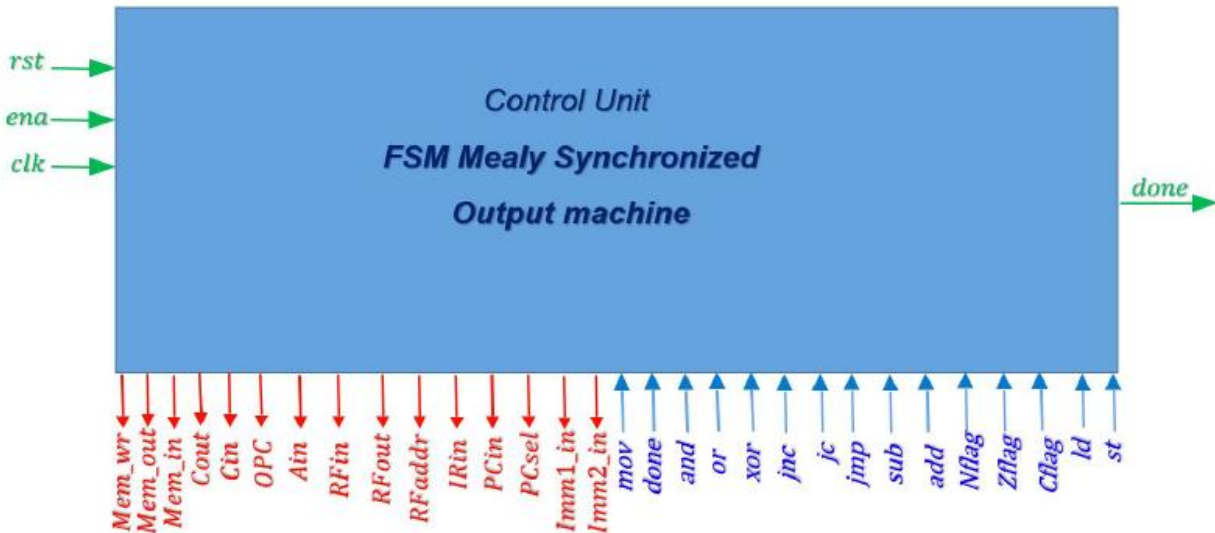
- Control Unit
- OPC Decoder
- Register File
- Program Memory
- Data Memory
- ALU
- TOP
- רכיבי מעטפת פנימיים ורכיבים מקשרים

תיאור המערכת:



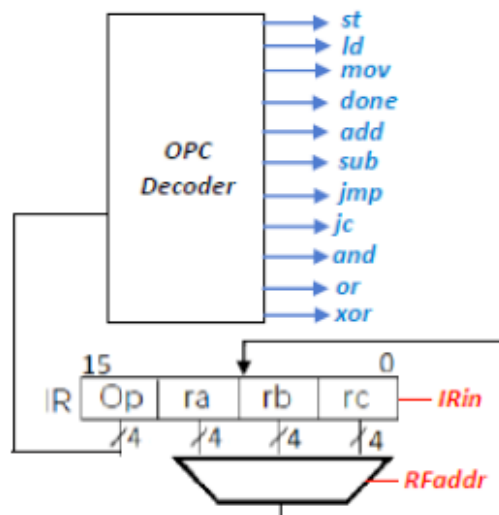
Control Unit

בארכיטקטורה המסופקת, יחידת הבקרה מקיימת אינטראקציה עם רכיבים שונים, כולל Program Memory, Data Memory, Register File, ALU, Memory. הוא משתמש במכונת מצב סופית (FSM) כדי לעבור בין מצבים שונים, כל אחד מתאים לשלב מסוים של ביצוע פקודות. יחידת הבקרה היא זו שפורטת כל שורה בתוכנית לכדי פעולות שהמערכת כולה, כל רכיב ותפקידו עבור אותה הוראה.



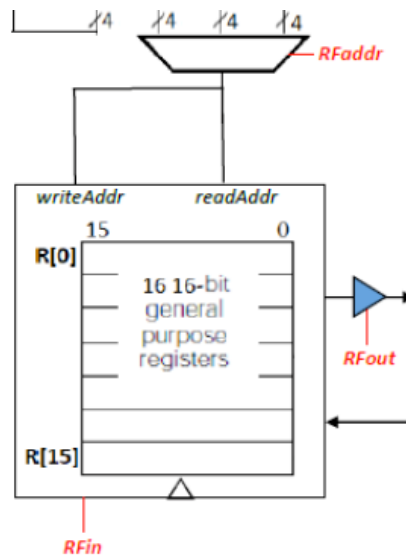
OPC Decoder

המטרה העיקרית של רכיב OPC Decoder היא לקחת קלט OpCode (מתוך הפקודה הנוכחית ב-IR) וליצור אותות בקרה עבור פעולות שונות המבוססות על קוד הפענוח. זהו חלק מכריע מיחידת הבקרה של המעבד, שכן הוא קובע איזו פעולה המעבד צריך לבצע לאחר מכן.



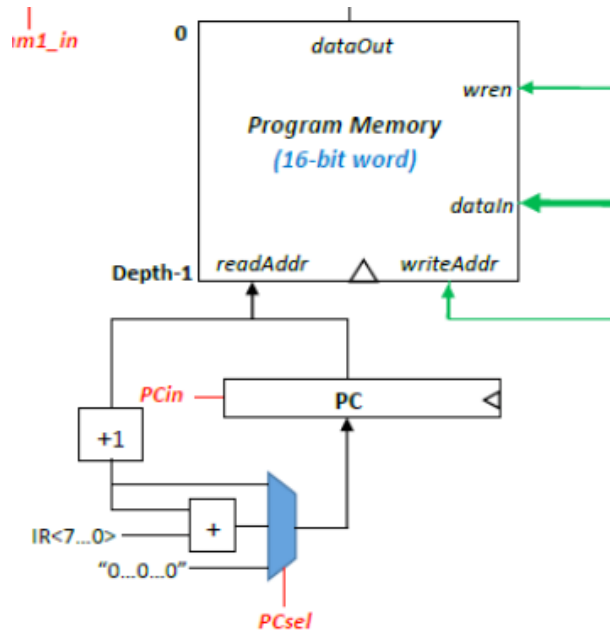
Register File

Register File בארכיטקטורת ה-CPU מאחסן נתוני ביניים ב-16 רגיסטרים למטרות כלליות, המאפשר טיפול יעיל בנתונים במהלך ביצוע פקודות. הוא מספק גישה מהירה לאופרנדים עבור ה-ALU ומקל על העברת נתונים מהירה אל וממנה באמצעות אותות 'RFinFromBus' ו-'RFoutToBus'. יחידת הבקרה מנהלת גישה רישום באמצעות אותות 'RFin', 'RFaddr', 'RFout' ו-'Rfaddr', ומבטיחה זרימת נתונים תקינה וביצוע הוראות. המטרה העיקרית של ה-RF היא לשפר את המהירות והיעילות של המעבד על ידי אספקת אחסון ואחזור מהירים של נתונים בשימוש תכוף.



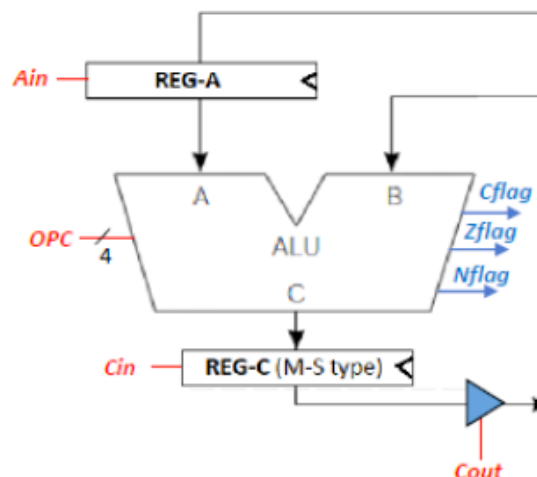
Program Memory

במעבד Multi-Cycle שתוכנן בגישת מכונת מצב סופית (FSM), Program Memory ממלא תפקיד מכריע באחסון ההוראות שהבקר מפרש ומוציא לפועל. מטרתו העיקרית היא לספק רצף הוראות, המובאות (Fetched) בדרך כלל בצורה מבוקרת בהתאם למצב הנוכחי של FSM. מבחינה פונקציונלית, זיכרון התוכנית מכיל את קוד המכונה או ההוראות הנחוצות למעבד לביצוע פעולות שהוגדרו על ידי התוכנית, מה שמקל על זרימת התוכנית והמעברים. זיכרון זה מבטיח שה-CPU פועל על ידי הזנת ההוראות הדרושות ב-IR.



ALU

במעבד Multi-Cycle שתוכנן בגישת מכונת מצב סופית (FSM), יחידת ה-ALU היא רכיב שאחראי על ביצוע פעולות אריתמטיות ולוגיות לפי הנחיות ה-FSM. ה-ALU מבצע חישובים כגון חיבור, חיסור ופעולות ביטים (bitwise) (AND, OR, XOR), בהתאם ל-opcode שסופק. מבחינה פונקציונלית, הוא מעבד אופרנדים של קלט, מייצר תוצאות ומעדכן דגלים לתוצאות Cflag, Zflag, Nflag, המשפיעות על זרימת הבקרה וקבלת ההחלטות בתוך ה-CPU. ה-FSM מתאם את ה-ALU על ידי אספקת אותות בקרה הקובעים את הפעולה הספציפית לביצוע במהלך כל מחזור, מה שמבטיח עיבוד מדויק ויעיל של ההוראות.



TOP

רכיב ה-TOP מייצג את המודול הראשי של מערכת מעבד Multi-Cycle, מתזמנת את זרימת הנתונים הכוללת ותיאום אותות בקרה לביצוע הוראות. המערכת בנויה עם מספר רכיבים מרכזיים, לרבות Program Memory, Data Memory, RF, ALU, ויחידת בקרה.

Top Entity כולל פרמטרים גנריים עבור רוחב נתונים, רוחב נרגיסטרים הכתובות, (AregWidth), ורוחב זיכרון הכתובות, (AmemWidth), מה שמתאים אותו לתצורות שונות. ה-outputs של ה-top כוללים שעון, איפוס (rst), enable (enb), ומספר אותות בקרה עבור פעולות זיכרון ותוכניות, (wEnaProg, wEnaMem, wAddrProg, wAddrMem, dataProg, dataMem).

יחידת הבקרה של המעבד מייצרת אותות בקרה ספציפיים, (Mem_wr, Mem_out, Mem_in, Cin, Cout, Ain, RFin, Rfout, RFaddr, IRin, PCin, PCsel, Imm1_in, Imm2_in, OPC) והמצב הנוכחי של ה-FSM. אותות אלה מתזמנים העברת נתונים בין רכיבים ופעולות בתוך ה-ALU, RF, והזיכרונות.

מצבי ה-FSM מכתיבים את רצף הפעולות לאחזור, פענוח, ביצוע ואחסון תוצאות של הוראות. לדוגמה, במהלך מצב האחזור, אוגר ההוראות (IRin) נטען עם ההוראה הבאה, ומונה התוכניות (PCin) מתעדכן. במצבי ביצוע, ה-ALU מבצע פעולות כמו חיבור, חיסור ופעולות לוגיות המבוססות על אות, OPC, והתוצאות נכתבות בחזרה לקובץ האוגר.

ה-TOP כולל גם רכיבי עזר לפענוח כתובות, טיפול בערכי immediate והתממשקות עם ה-testbench (TBactive). רכיבים אלו מבטיחים שה-CPU פועל בצורה נכונה וניתן לבדיקה יעילה.

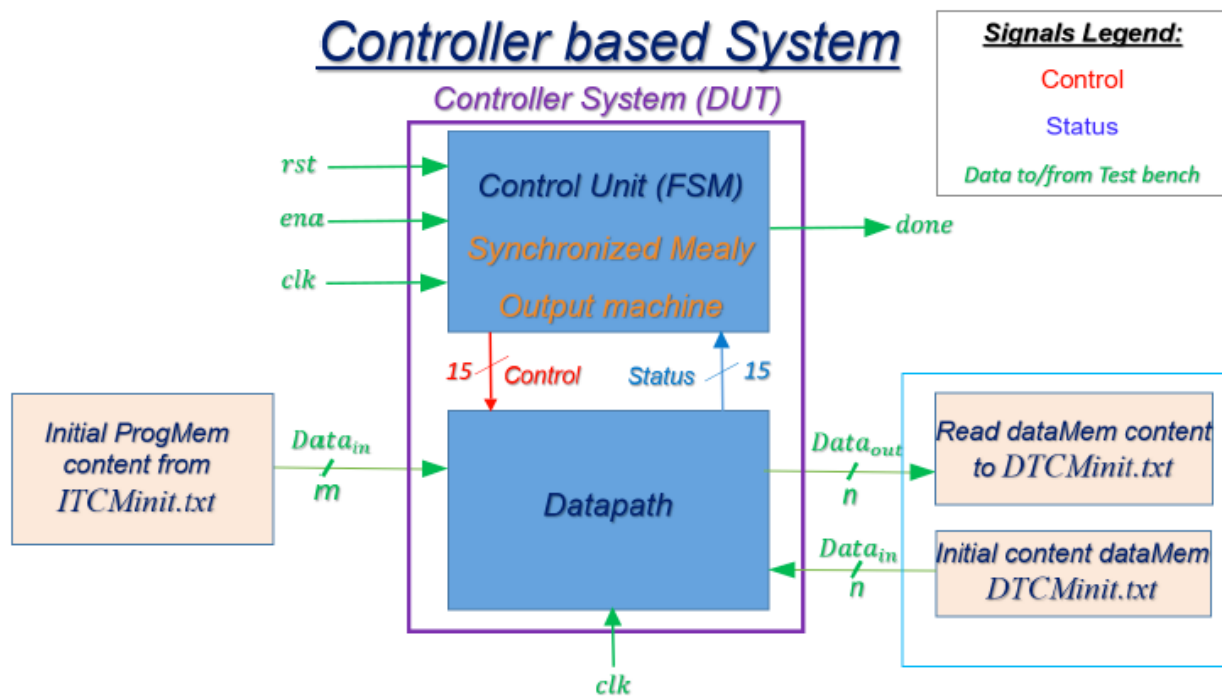
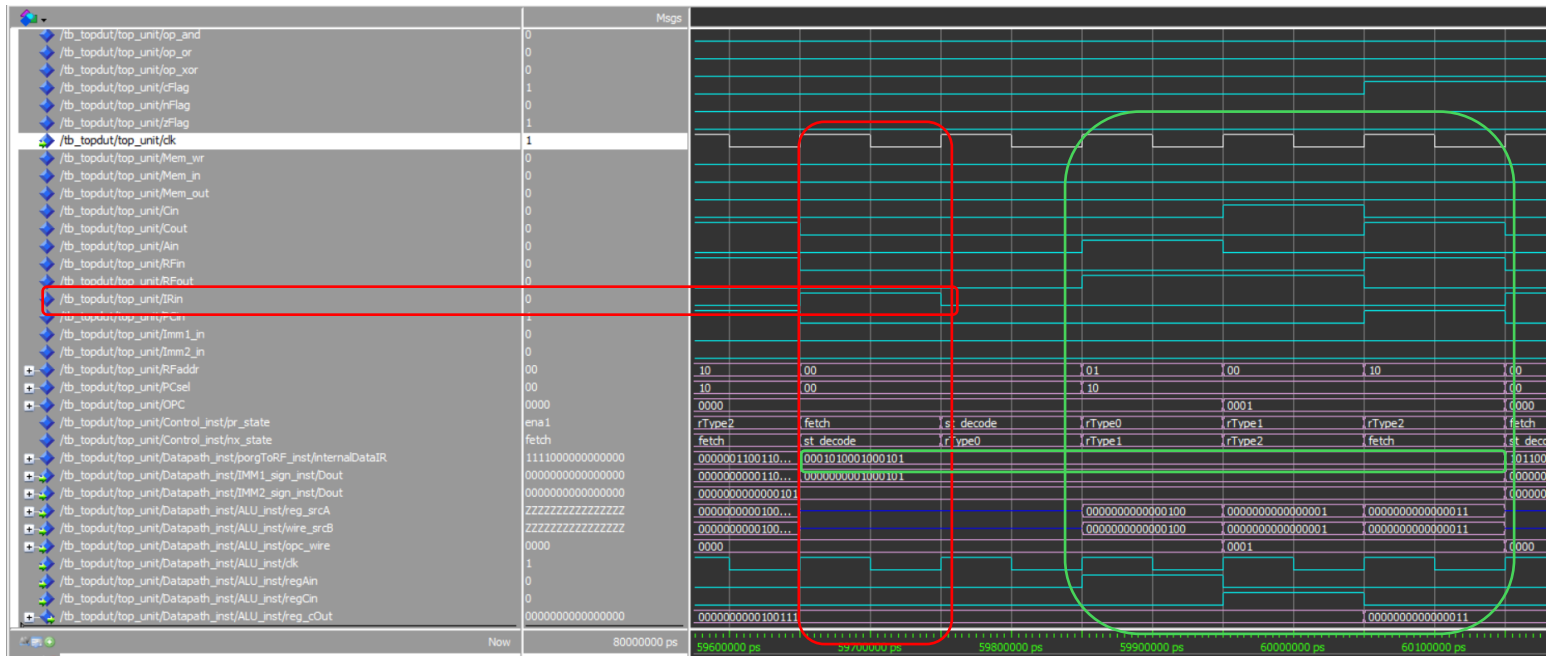


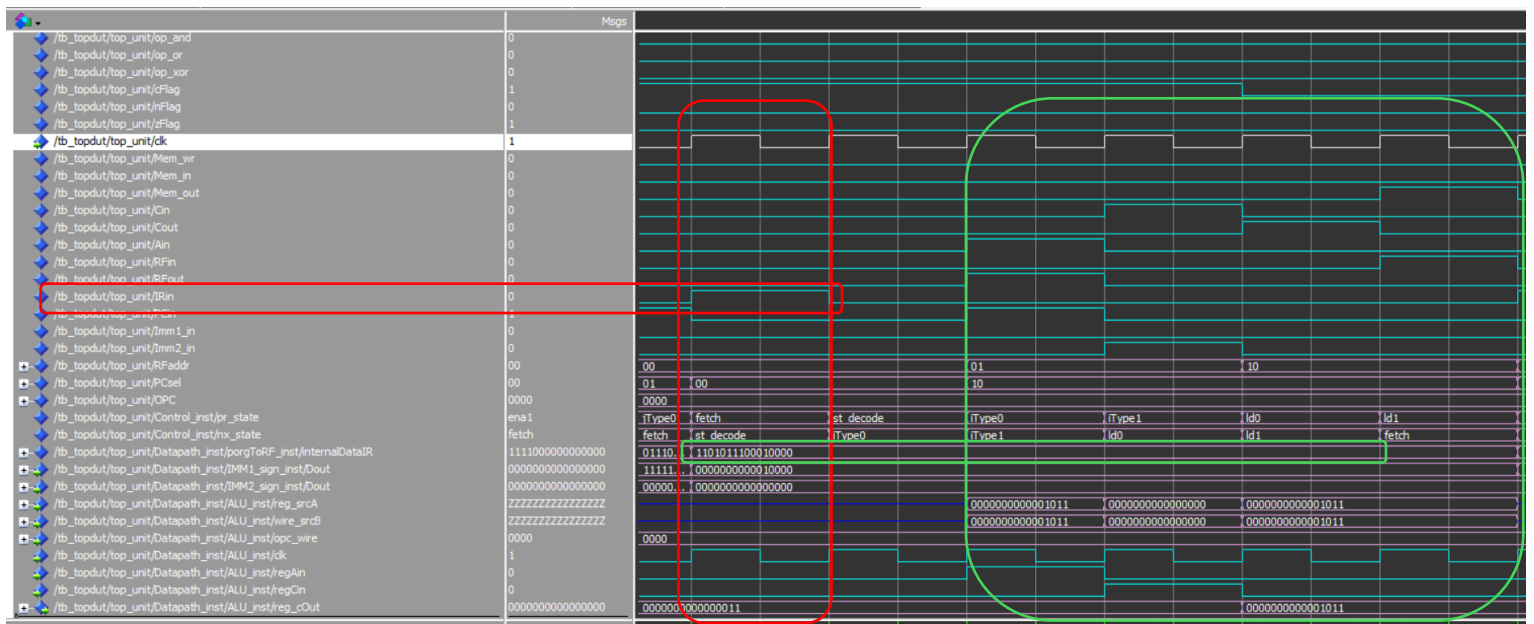
Figure 1: Overall DUT structure

מודל סים עם הסברים :



באדום : שלב fetch בו ניתן לראות שמתקבל $irin=1$ כלומר מבוצע השלב הראשון ב-FSM שלנו, ניתן לראות שבשלב זה הערכים מאופסים.
 בירוק : אחרי שלב decode שנדרש בגלל מגבלות זמנים של לוגיקה בפינוח של ה opcode ולכן לא ניתן לבצע את הפענוח באותו סייקל של fetch. אפשר לראות שמתקבלת פקודת rtype לפי ה IR שמבקש לבצע sub 0001.

דוגמא נוספת :



באדום : שלב fetch בו ניתן לראות שמתקבל $irin=1$ כלומר מבוצע השלב הראשון בFSM שלנו, ניתן לראות שבשלב זה הערכים מאופסים.

בירוק : ניתן לראות שה- ir דורש פקודת ld מכיוון שה opcode 1101 ואפשר לראות את שלבי ה states לפי ה FSM

בנוסף ניתן לראות ב srcA ו srcB את הערכים הנכונים של הרגיסטרים אליהם צריך לקרוא ומהם צריך לכתוב שמגיעים מהBUS.