

CPU Architecture

LAB 5 assignment

FPGA based Digital Design

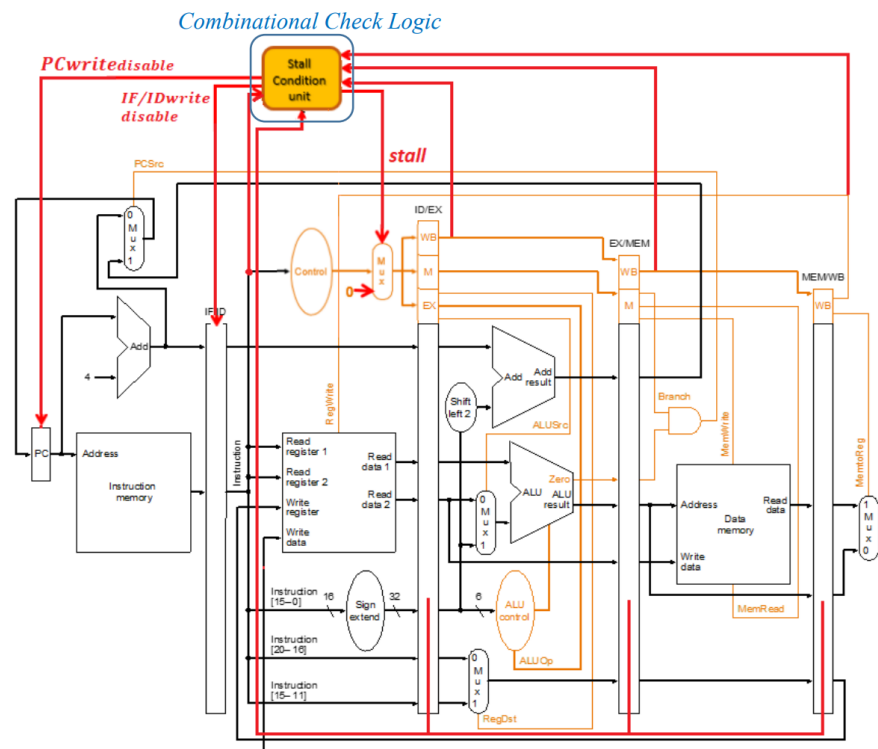
מגישים:

יבגני יגודין- 324432988

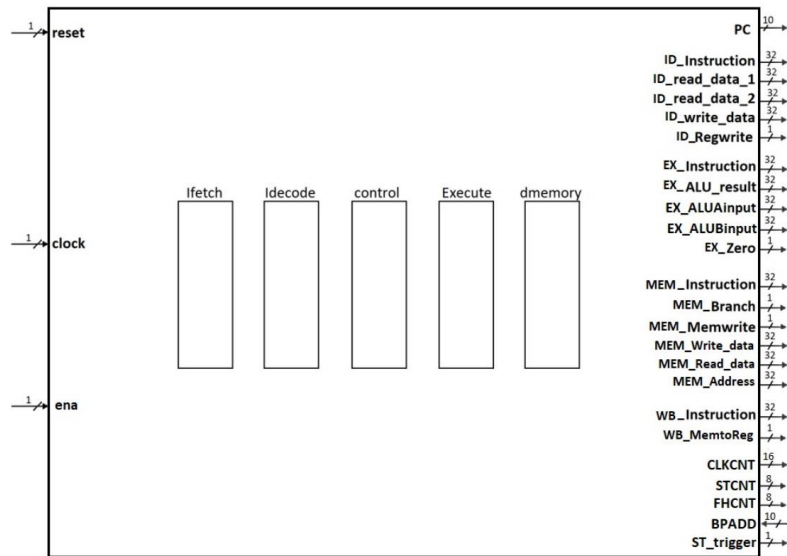
אור יעקבי- 206827164

:TOP LEVEL BLOCK

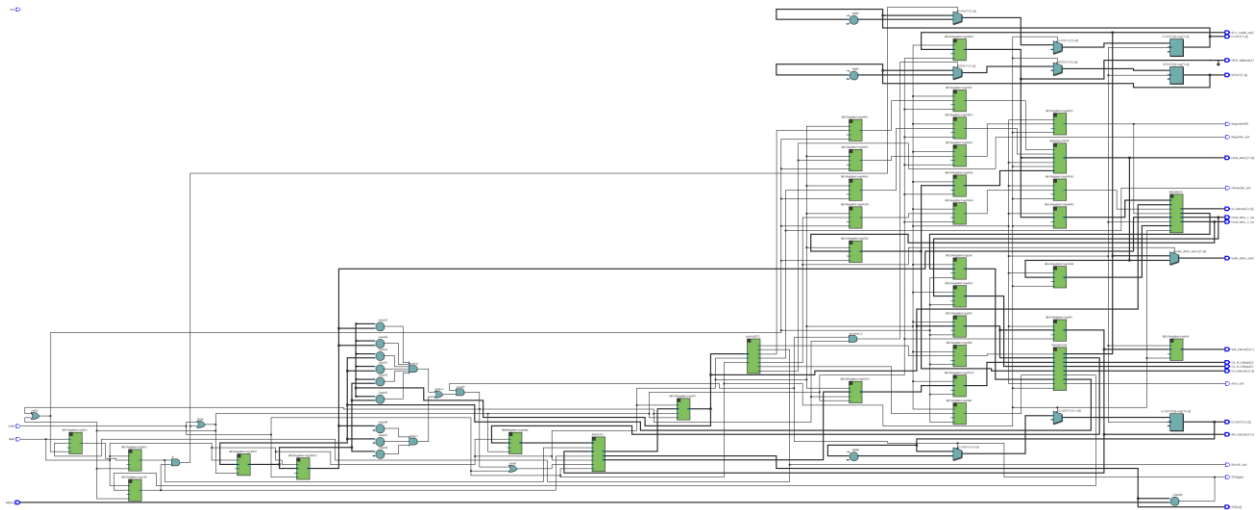
הצלחנו לממש את כל הפקודות , שידרגנו את המערכת ל- PIPELINED , מימשנו FLUSH וגם HAZARD
DETECTION ובפרט- STALL . (לא הצלחנו לממש את FORWARDING לצעריינו)



ה-TOP שלנו בנוי ממש כמו בשרטוט הבא עם הפלטים והקלטים הבאים:



ה-RTL שלו נראה ככה:



תוצאות הקימפול:

Flow Summary



<<Filter>>

Flow Status	Successful - Sun Jun 18 05:35:32 2023
Quartus Prime Version	21.1.0 Build 842 10/21/2021 SJ Lite Edition
Revision Name	laba5
Top-level Entity Name	MIPS
Family	Cyclone V
Device	5CSXFC6D6F31C6
Timing Models	Final
Logic utilization (in ALMs)	1,452 / 41,910 (3 %)
Total registers	1399
Total pins	1 / 499 (< 1 %)
Total virtual pins	444
Total block memory bits	16,384 / 5,662,720 (< 1 %)
Total DSP Blocks	2 / 112 (2 %)
Total HSSI RX PCSs	0 / 9 (0 %)
Total HSSI PMA RX Deserializers	0 / 9 (0 %)
Total HSSI TX PCSs	0 / 9 (0 %)
Total HSSI PMA TX Serializers	0 / 9 (0 %)
Total PLLs	0 / 15 (0 %)
Total DLLs	0 / 4 (0 %)

מסלול קריטי ותדר מקסימלי:

התדר המקסימלי כפי שהסימולציה מראה הינו 80 מגה הרץ.

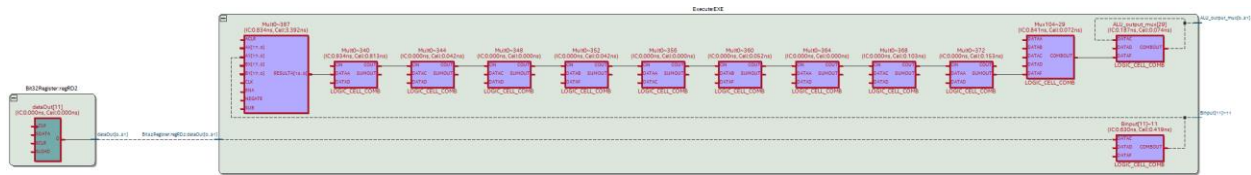
Slow 1100mV 85C Model Fmax Summary



<<Filter>>

	Fmax	Restricted Fmax	Clock Name	Note
1	36.37 MHz	36.37 MHz	Bit32Register:regExt[dataOut[0]	
2	80.35 MHz	80.35 MHz	clock	

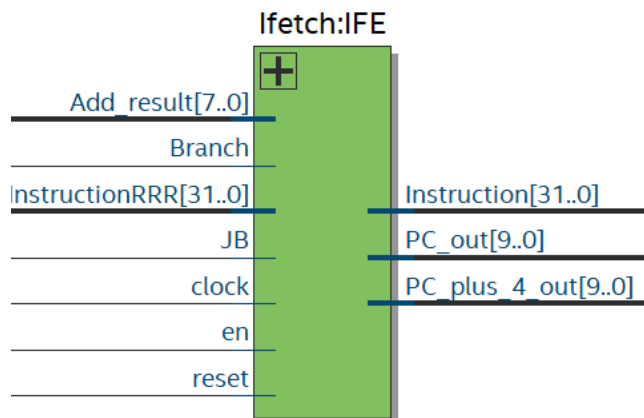
המסלול הקריטי עובר דרך יחידת execute ובפרט דרך המכפל.
 זה לא מפתיה שהמסלול עובר דרך מכפל כי המכפל הוא יחידה ממש כבדה בסיבוכיות ריצה בגלל המימוש הטורי שלה.



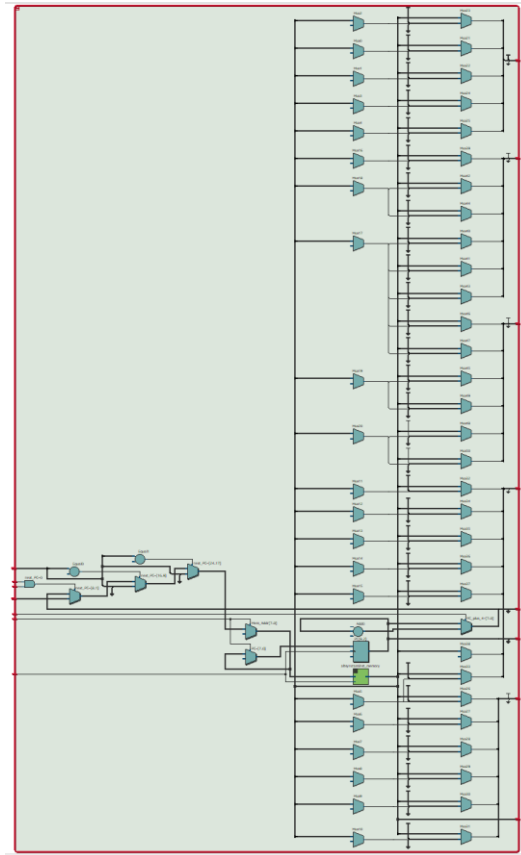
נדבר על כל תת בלוק בנפרד:

FETCH

הרכיב מקדם את ה-PC וקורא מן קובץ מקור את רשימת פקודות או קוד שעליו לבצע.
 את הפקודה שקרא, מעבר לשאר המערכת בצורה טורית.
 כמו כן יש בו מנגנון של הסתעפויות.

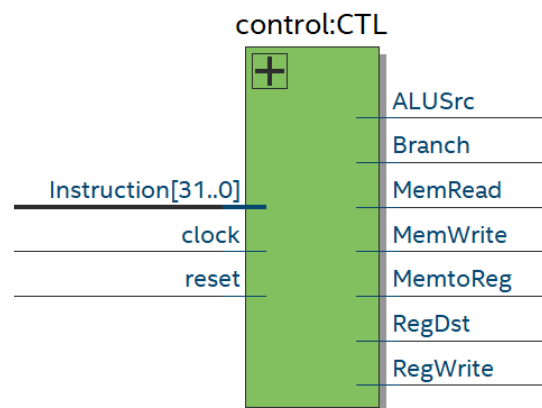


RTL-ה

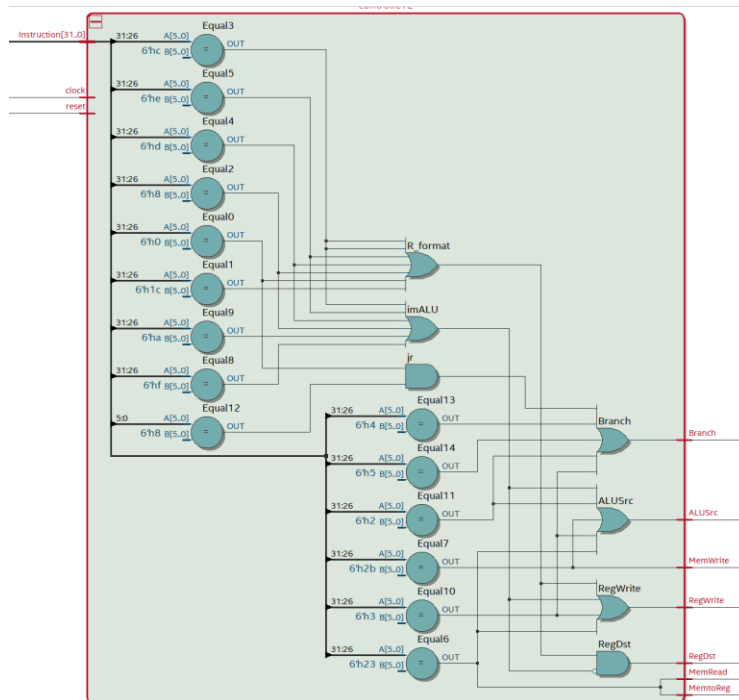


CONTROL

היחידה מקבלת את הפקודה ובהתאמה מפעילה קווי בקרה מתאימים בכל שכבה בנפרד. למשל פותחת אופציה לקרוא או לכתוב מזכרון או מרגיסטרים.

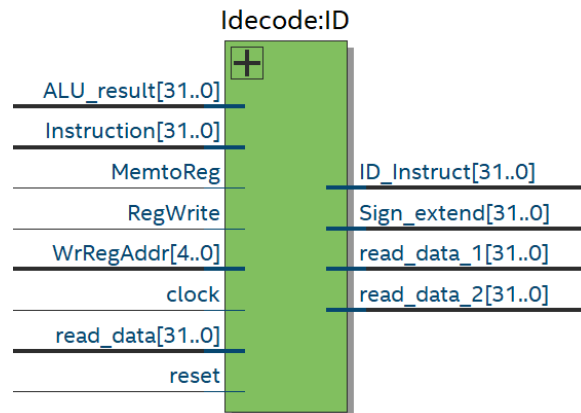


RTL-ה

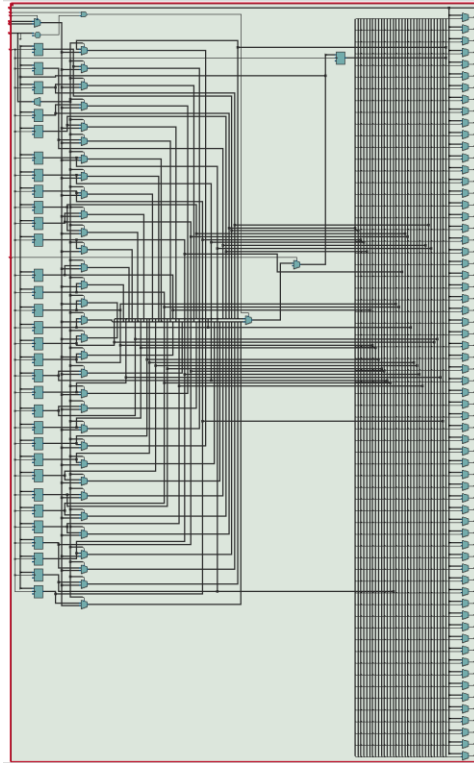


DECODER

היחידה מקבלת את הפקודה מ-IFETCH ובהתאמה מפרשת אילו רגיסטרים הם מקור, מטרה ויעד ובהתאמה פותחת אותם לקריאה ולכתיבה אם מדובר ברגיסטר יעד. כל זה תוך כדי הקשבה לקווי בקרה של CONTROL ועוד.

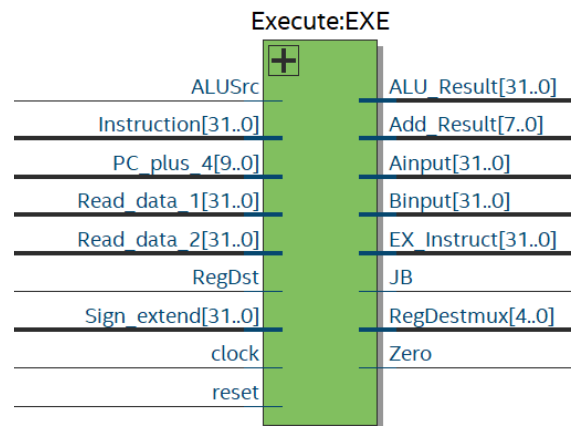


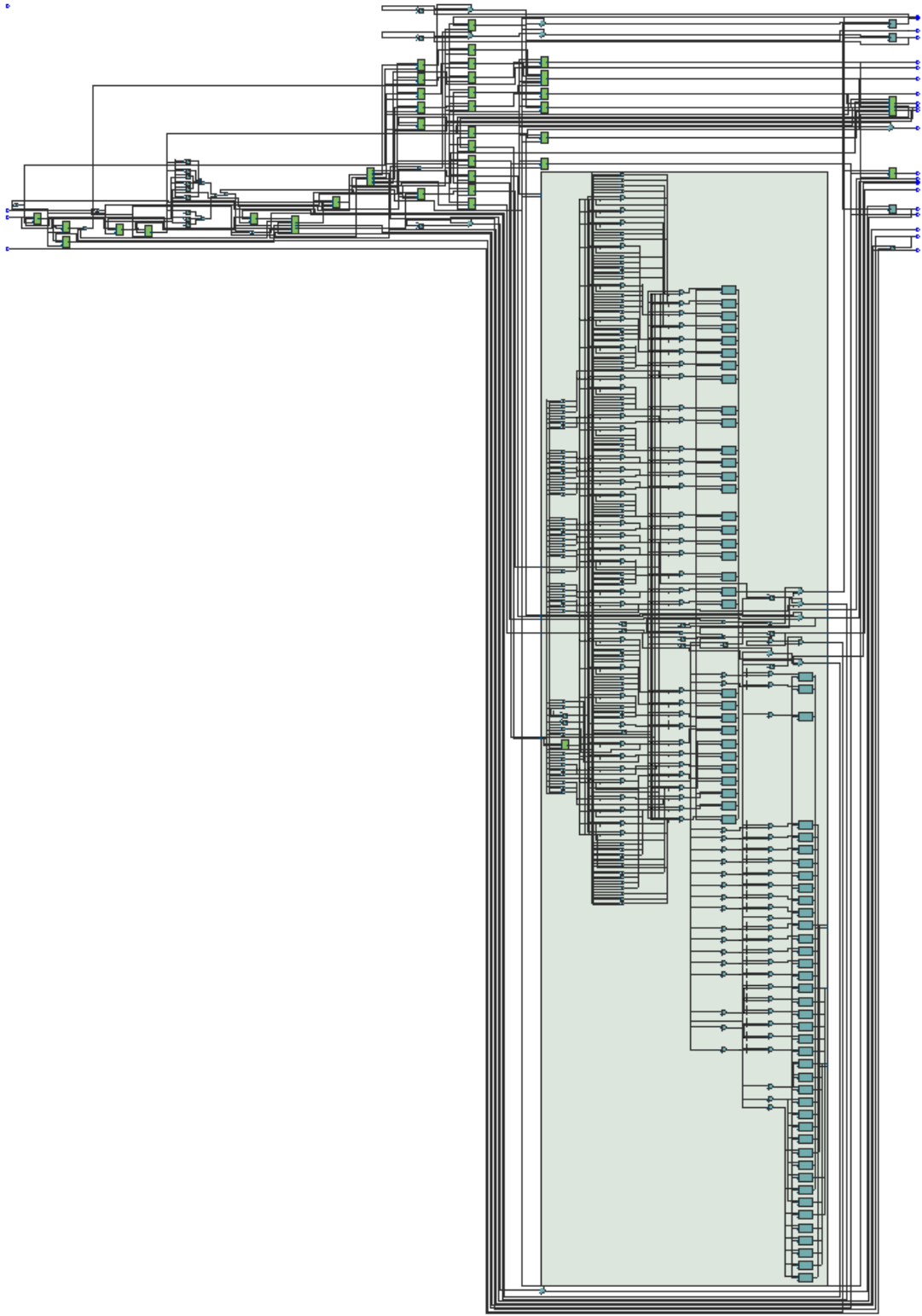
ה-RTL:



EXECUTE

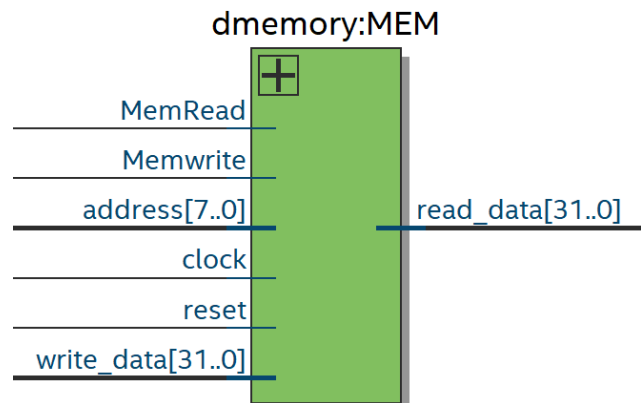
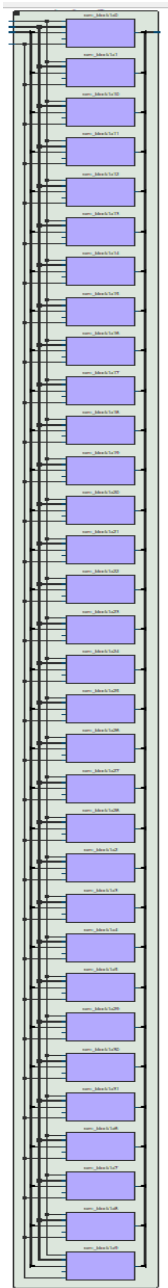
היחידה מבצעת פעולות מתמטיות לפי הפקודה שקיבלה ו-DECODER. ה-DECODER מספק אותה בתוכן הרגיסטרים שעליהם יבוצעו הפעולות. היחידה גם מחשבים האם יקרו חלק מההסתעפויות (למשל BEQ/BNE)





DATA MEMORY

היחידה טוענת את הזכרון ההתחלתי ועובדת איתו במהלך הקוד.
ניתן לקרוא ממנה וגם לכתוב אליה במהלך הריצה או בסוף הריצה לפי הצורך.
השליטה בכתיבה/קריאה/כתובת מתבצעת דרך CONTROL ו-תוצאת EXECUTE.

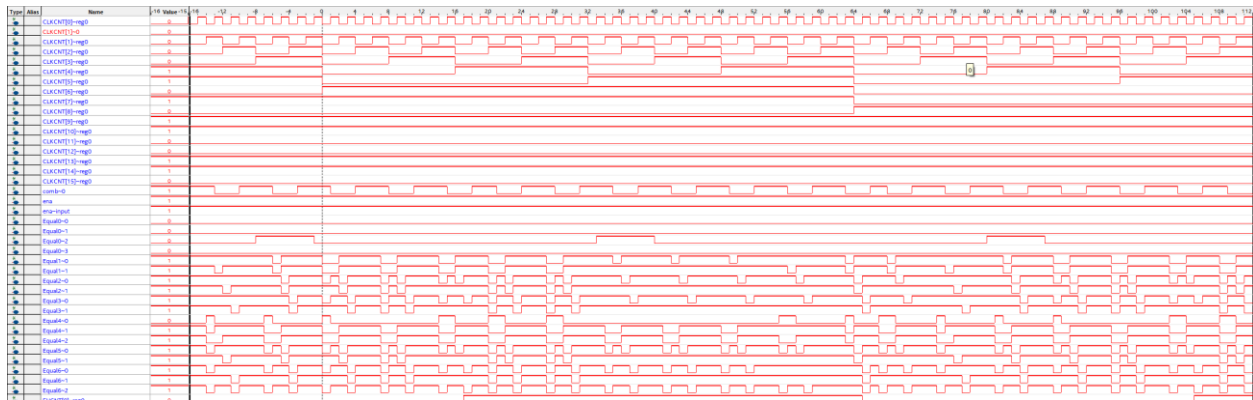


RTL-ה

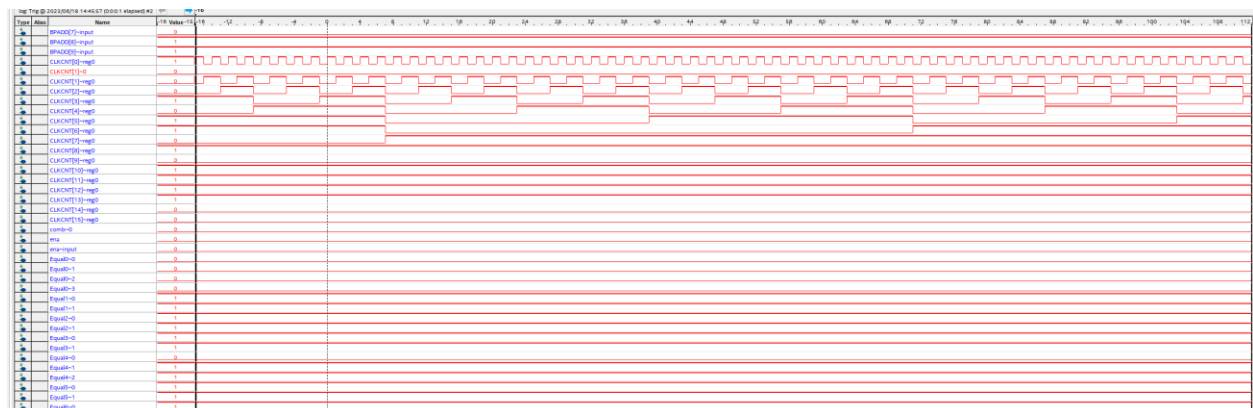
SINGAL TAP

לאחר צריבה של קודה ל-FPGA הרצנו את ה-SIGNAL TAP.

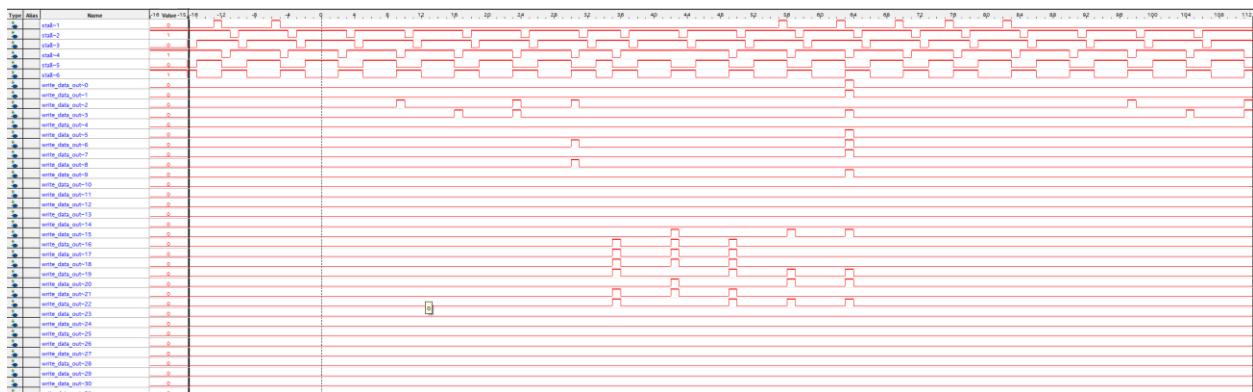
בריצה רגילה נראה את הקווים עובדים, למעלה השעון רץ ו-CLKCNT עובד כ-COUNTER כצפוי.



בתמונה הבאה רואים את אותו דבר אבל זמן שמתג 9-SW סגור. כלומר ה-ENA מכובה ולכן הקווים שתקו, חוץ מהשעון שהוא כמובן פועם



בתמונה הבאה רואים את המשך של תמונה ראשונה, פשוט קווים אחרים.



בתמונה הבאה רואים את המערכת כש-KEO לחוץ כלומר יש RESET קבוע.

File Type ID: 202007011427200001_ajpaword-01			19
Type	Index	Name	Value
★	0	CLACNT02-mq0	0
★	1	CLACNT02-mq1	0
★	2	CLACNT02-mq2	0
★	3	CLACNT02-mq3	0
★	4	CLACNT02-mq4	0
★	5	CLACNT02-mq5	0
★	6	CLACNT02-mq6	0
★	7	CLACNT02-mq7	0
★	8	CLACNT02-mq8	0
★	9	CLACNT02-mq9	0
★	10	CLACNT02-mq10	0
★	11	CLACNT02-mq11	0
★	12	CLACNT02-mq12	0
★	13	CLACNT02-mq13	0
★	14	CLACNT02-mq14	0
★	15	CLACNT02-mq15	0
★	16	CLACNT02-mq16	0
★	17	CLACNT02-mq17	0
★	18	CLACNT02-mq18	0
★	19	CLACNT02-mq19	0
★	20	CLACNT02-mq20	0
★	21	CLACNT02-mq21	0
★	22	CLACNT02-mq22	0
★	23	CLACNT02-mq23	0
★	24	CLACNT02-mq24	0
★	25	CLACNT02-mq25	0
★	26	CLACNT02-mq26	0
★	27	CLACNT02-mq27	0
★	28	CLACNT02-mq28	0
★	29	CLACNT02-mq29	0
★	30	CLACNT02-mq30	0
★	31	CLACNT02-mq31	0
★	32	CLACNT02-mq32	0
★	33	CLACNT02-mq33	0
★	34	CLACNT02-mq34	0
★	35	CLACNT02-mq35	0
★	36	CLACNT02-mq36	0
★	37	CLACNT02-mq37	0
★	38	CLACNT02-mq38	0
★	39	CLACNT02-mq39	0
★	40	CLACNT02-mq40	0
★	41	CLACNT02-mq41	0
★	42	CLACNT02-mq42	0
★	43	CLACNT02-mq43	0
★	44	CLACNT02-mq44	0
★	45	CLACNT02-mq45	0
★	46	CLACNT02-mq46	0
★	47	CLACNT02-mq47	0
★	48	CLACNT02-mq48	0
★	49	CLACNT02-mq49	0
★	50	CLACNT02-mq50	0
★	51	CLACNT02-mq51	0
★	52	CLACNT02-mq52	0
★	53	CLACNT02-mq53	0
★	54	CLACNT02-mq54	0
★	55	CLACNT02-mq55	0
★	56	CLACNT02-mq56	0
★	57	CLACNT02-mq57	0
★	58	CLACNT02-mq58	0
★	59	CLACNT02-mq59	0
★	60	CLACNT02-mq60	0
★	61	CLACNT02-mq61	0
★	62	CLACNT02-mq62	0
★	63	CLACNT02-mq63	0
★	64	CLACNT02-mq64	0
★	65	CLACNT02-mq65	0
★	66	CLACNT02-mq66	0
★	67	CLACNT02-mq67	0
★	68	CLACNT02-mq68	0
★	69	CLACNT02-mq69	0
★	70	CLACNT02-mq70	0
★	71	CLACNT02-mq71	0
★	72	CLACNT02-mq72	0
★	73	CLACNT02-mq73	0
★	74	CLACNT02-mq74	0
★	75	CLACNT02-mq75	0
★	76	CLACNT02-mq76	0
★	77	CLACNT02-mq77	0
★	78	CLACNT02-mq78	0
★	79	CLACNT02-mq79	0
★	80	CLACNT02-mq80	0
★	81	CLACNT02-mq81	0
★	82	CLACNT02-mq82	0
★	83	CLACNT02-mq83	0
★	84	CLACNT02-mq84	0
★	85	CLACNT02-mq85	0
★	86	CLACNT02-mq86	0
★	87	CLACNT02-mq87	0
★	88	CLACNT02-mq88	0
★	89	CLACNT02-mq89	0
★	90	CLACNT02-mq90	0
★	91	CLACNT02-mq91	0
★	92	CLACNT02-mq92	0
★	93	CLACNT02-mq93	0
★	94	CLACNT02-mq94	0
★	95	CLACNT02-mq95	0
★	96	CLACNT02-mq96	0
★	97	CLACNT02-mq97	0
★	98	CLACNT02-mq98	0
★	99	CLACNT02-mq99	0
★	100	CLACNT02-mq100	0
★	101	CLACNT02-mq101	0
★	102	CLACNT02-mq102	0
★	103	CLACNT02-mq103	0
★	104	CLACNT02-mq104	0
★	105	CLACNT02-mq105	0
★	106	CLACNT02-mq106	0
★	107	CLACNT02-mq107	0
★	108	CLACNT02-mq108	0
★	109	CLACNT02-mq109	0
★	110	CLACNT02-mq110	0
★	111	CLACNT02-mq111	0
★	112	CLACNT02-mq112	0
★	113	CLACNT02-mq113	0
★	114	CLACNT02-mq114	0
★	115	CLACNT02-mq115	0
★	116	CLACNT02-mq116	0
★	117	CLACNT02-mq117	0
★	118	CLACNT02-mq118	0
★	119	CLACNT02-mq119	0
★	120	CLACNT02-mq120	0
★	121	CLACNT02-mq121	0
★	122	CLACNT02-mq122	0
★	123	CLACNT02-mq123	0
★	124	CLACNT02-mq124	0
★	125	CLACNT02-mq125	0
★	126	CLACNT02-mq126	0
★	127	CLACNT02-mq127	0
★	128	CLACNT02-mq128	0
★	129	CLACNT02-mq129	0
★	130	CLACNT02-mq130	0
★	131	CLACNT02-mq131	0
★	132	CLACNT02-mq132	0
★	133	CLACNT02-mq133	0
★	134	CLACNT02-mq134	0
★	135	CLACNT02-mq135	0
★	136	CLACNT02-mq136	0
★	137	CLACNT02-mq137	0
★	138	CLACNT02-mq138	0
★	139	CLACNT02-mq139	0
★	140	CLACNT02-mq140	0
★	141	CLACNT02-mq141	0
★	142	CLACNT02-mq142	0
★	143	CLACNT02-mq143	0
★	144	CLACNT02-mq144	0
★	145	CLACNT02-mq145	0
★	146	CLACNT02-mq146	0
★	147	CLACNT02-mq147	0
★	148	CLACNT02-mq148	0
★	149	CLACNT02-mq149	0
★	150	CLACNT02-mq150	0
★	151	CLACNT02-mq151	0
★	152	CLACNT02-mq152	0
★	153	CLACNT02-mq153	0
★	154	CLACNT02-mq154	0
★	155	CLACNT02-mq155	0
★	156	CLACNT02-mq156	0
★	157	CLACNT02-mq157	0
★	158	CLACNT02-mq158	0
★	159	CLACNT02-mq159	0
★	160	CLACNT02-mq160	0
★	161	CLACNT02-mq161	0
★	162	CLACNT02-mq162	0
★	163	CLACNT02-mq163	0
★	164	CLACNT02-mq164	0
★	165	CLACNT02-mq165	0
★	166	CLACNT02-mq166	0
★	167	CLACNT02-mq167	0
★	168	CLACNT02-mq168	0
★	169	CLACNT02-mq169	0
★	170	CLACNT02-mq170	0
★	171	CLACNT02-mq171	0
★	172	CLACNT02-mq172	0
★	173	CLACNT02-mq173	0
★	174	CLACNT02-mq174	0
★	175	CLACNT02-mq175	0
★	176	CLACNT02-mq176	0
★	177	CLACNT02-mq177	0
★	178	CLACNT02-mq178	0
★	179	CLACNT02-mq179	0
★	180	CLACNT02-mq180	0
★	181	CLACNT02-mq181	0
★	182	CLACNT02-mq182	0
★	183	CLACNT02-mq183	0
★	184	CLACNT02-mq184	0
★	185	CLACNT02-mq185	0
★	186	CLACNT02-mq186	0
★	187	CLACNT02-mq187	0
★	188	CLACNT02-mq188	0
★	189	CLACNT02-mq189	0
★	190	CLACNT02-mq190	0
★	191	CLACNT02-mq191	0
★	192	CLACNT02-mq192	0
★	193	CLACNT02-mq193	0
★	194	CLACNT02-mq194	0
★	195	CLACNT02-mq195	0
★	196	CLACNT02-mq196	0
★	197	CLACNT02-mq197	0
★	198	CLACNT02-mq198	0
★	199	CLACNT02-mq199	0
★	200	CLACNT02-mq200	0
★	201	CLACNT02-mq201	0
★	202	CLACNT02-mq202	0
★	203	CLACNT02-mq203	0
★	204	CLACNT02-mq204	0
★	205	CLACNT02-mq205	0
★	206	CLACNT02-mq206	0
★	207	CLACNT02-mq207	0
★	208	CLACNT02-mq208	0
★	209	CLACNT02-mq209	0
★	210	CLACNT02-mq210	0
★	211	CLACNT02-mq211	0
★	212	CLACNT02-mq212	0
★	213	CLACNT02-mq213	0
★	214	CLACNT02-mq214	0
★	215	CLACNT02-mq215	0
★	216	CLACNT02-mq216	0
★	217	CLACNT02-mq217	0
★	218	CLACNT02-mq218	0
★	219	CLACNT02-mq219	0
★	220	CLACNT02-mq220	0
★	221	CLACNT02-mq221	0
★	222	CLACNT02-mq222	0
★	223	CLACNT02-mq223	0
★	224	CLACNT02-mq224	0
★	225	CLACNT02-mq225	0
★	226	CLACNT02-mq226	0
★	227	CLACNT02-mq227	0
★	228	CLACNT02-mq228	0
★	229	CLACNT02-mq229	0
★	230	CLACNT02-mq230	0
★	231	CLACNT02-mq231	0
★	232	CLACNT02-mq232	0
★	233	CLACNT02-mq233	0
★	234	CLACNT02-mq234	0
★	235	CLACNT02-mq235	0
★	236	CLACNT02-mq236	0
★	237	CLACNT02-mq237	0
★	238	CLACNT02-mq238	0
★	239	CLACNT02-mq239	0
★	240	CLACNT02-mq240	0
★	241	CLACNT02-mq241	0
★	242	CLACNT02-mq242	0
★	243	CLACNT02-mq243	0
★	244	CLACNT02-mq244	0
★	245	CLACNT02-mq245	0
★	246	CLACNT02-mq246	0
★	247	CLACNT02-mq247	0
★	248	CLACNT02-mq248	0
★	249	CLACNT02-mq249	0
★	250	CLACNT02-mq250	0
★	251	CLACNT02-mq251	0
★	252	CLACNT02-mq252	0
★	253	CLACNT02-mq253	0
★	254	CLACNT02-mq254	0
★	255	CLACNT02-mq255	0
★	256	CLACNT02-mq256	0
★	257	CLACNT02-mq257	0
★	258	CLACNT02-mq258	0
★	259	CLACNT02-mq259	0
★	260	CLACNT02-mq260	0
★	261	CLACNT02-mq261	0
★	262	CLACNT02-mq262	0
★	263	CLACNT02-mq263	0
★	264	CLACNT02-mq264	0
★	265	CLACNT02-mq265	0
★	266	CLACNT02-mq266	0
★	267	CLACNT02-mq267	0
★	268	CLACNT02-mq268	0
★	269	CLACNT02-mq269	0
★	270	CLACNT02-mq270	0
★	271	CLACNT02-mq271	0
★	272	CLACNT02-mq272	0
★	273	CLACNT02-mq273	0
★	274	CLACNT02-mq274	0
★	275	CLACNT02-mq275	0
★	276	CLACNT02-mq276	0
★	277	CLACNT02-mq277	0
★	278	CLACNT02-mq278	0
★	279	CLACNT02-mq279	0
★	280	CLACNT02-mq280	0
★	281	CLACNT02-mq281	0
★	282	CLACNT02-mq282	0
★	283	CLACNT02-mq283	0
★	284	CLACNT02-mq284	0
★	285	CLACNT02-mq285	0
★	286	CLACNT02-mq286	0
★	287	CLACNT02-mq287	0
★	288	CLACNT02-mq288	0
★	289	CLACNT02-mq289	0
★	290	CLACNT02-mq290	0
★	291	CLACNT02-mq291	0
★	292	CLACNT02-mq292	0
★	293	CLACNT02-mq293	0
★	294	CLACNT02-mq294	0
★	295	CLACNT02-mq295	0
★	296	CLACNT02-mq296	0
★	297	CLACNT02-mq297	0
★	298	CLACNT02-mq298	0
★	299	CLACNT02-mq299	0
★	300	CLACNT02-mq300	0
★	301	CLACNT02-mq301	0
★	302	CLACNT02-mq302	0
★	303	CLACNT02-mq303	0
★	304	CLACNT02-mq304	0
★	305	CLACNT02-mq305	0
★	306	CLACNT02-mq306	0
★	307	CLACNT02-mq307	0
★	308	CLACNT02-mq308	0
★	309	CLACNT02-mq309	0
★	310	CLACNT02-mq310	0
★	311	CLACNT02-mq311	0
★	312	CLACNT02-mq312	0
★	313	CLACNT02-mq313	0
★	314	CLACNT02-mq314	0
★	315	CLACNT02-mq315	0
★	316	CLACNT02-mq316	0
★	317	CLACNT02-mq317	0
★	318	CLACNT02-mq318	0
★	319	CLACNT02-mq319	0
★	320	CLACNT02-mq320	0
★			

קוד מטריצות:

```
.data
Mat1: .word 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16
Mat2: .word 13, 14, 15, 16, 9, 10, 11, 12, 5, 6, 7, 8, 1, 2, 3, 4
resMat: .word 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0

.text

addMats:
    addi $t4, $zero, 0           # Initialize the row counter

loop:
    move $t6, $t4                # Calculate the index of the current element
    sll $t6, $t6, 2              # Scale the index by 4 (word size)

    lw $t7, Mat1($t6)            # Load element from Mat1
    lw $t8, Mat2($t6)            # Load element from Mat2
    add $t9, $t7, $t8             # Add the elements

    sw $t9, resMat($t6)          # Store the result in resMat

    addi $t4, $t4, 1             # Increment the row counter
    slti $t6, $t4, 16            # Check if the row counter < N
    bne $t6, $zero, loop
    addi $zero, $zero, 0
    jr $ra                       # Return
```

השוואת תוכן הזיכרון נתונים :

Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)	Value (+24)	Value (+28)
0	1	2	3	4	5	6	7	8
32	9	10	11	12	13	14	15	16
64	13	14	15	16	9	10	11	12
96	5	6	7	8	1	2	3	4
128	14	16	18	20	14	16	18	20
160	14	16	18	20	14	16	18	20

48	20	47	18	46	16	45	14	44	20	43	18	42	16	41	14	40	20	39	18	38	16	37	14	36	20	35	18	34	16	33	14	32	4	31	3	30	2	29	1	28	8	27	7	26	6	25	5	24	12	23	11	22	10	21	9	20	16	19	15	18	14	17	13	16	16	15	15	14	14	13	13	12	12	11	11	10	10	9	9	8	8	7	7	6	6	5	5	4	4	3	3	2	2	1	2
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	---	----	----	----	----	----	----	----	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

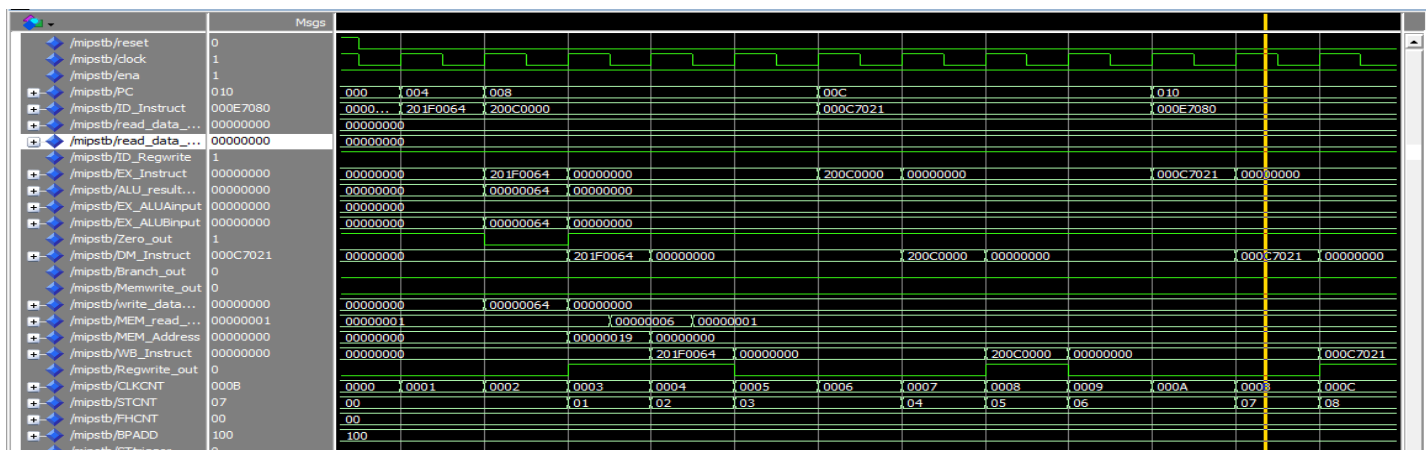
ניתן לראות שתוכן מטריצת התוצאה זהה.

השוואת תוכן הרגיסטרים

Memory Data - /mipstb/L1/ID/register_array - Default		
0	0	
1	0	
2	0	
3	0	
4	0	
5	0	
6	0	
7	0	
8	0	
9	0	
10	0	
11	0	
12	16	
13	0	
14	0	
15	16	
16	0	
17	0	
18	0	
19	0	
20	0	
21	0	
22	0	
23	0	
24	4	
25	20	
26	0	
27	0	
28	0	
29	0	
30	0	
31	100	

Number	Value
0	0
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	16
13	0
14	0
15	16
16	0
17	0
18	0
19	0
20	0
21	0
22	0
23	0
24	4
25	20
26	0
27	0
28	6144
29	12284
30	0
31	100
	12340
	0
	0

Wave form:



גם כאן ניתן לראות את הצורה האלכסונית של דחיפת הפקודות ב פיפליין, ספירת הסטולים והפלאש.

בסוף התכנית:

+ /mipstb/CLKCNT	01EB
+ /mipstb/STCNT	26
+ /mipstb/FHCNT	10

$$IPC = \frac{CLKCNT - (STCNT + 4 + FHCNT * dept)}{CLKCNT} = \frac{491 - (26 + 4 + 10 * 3)}{491} = 0.87$$

נשים לב שכמות הסטולים נמוכה מכיוון שאין לנו בקוד פקודות קריאה- כתיבה ואין סטולים כי אין הרבה קפיצות לכן קיבלנו תוצאה יחסית גבוהה.

דוגמה לקוד שלנו – QA (הקבצים של DO) שייכים לקוד הזה.

נדגים את הקוד שלנו שכתבנו, הוא מכיל את כל הסוגים של הסתעפויות ותלויות בין רגיסטרים. ונציג את תוכן הרגיסטרים בסיום הריצה לפי MARS ולפי סימולציית MODELSIM.

בצד ימין, תוכן הרגיסטרים לפי MARS, באמצע לפי MODELSIM ובצד שמאל הקוד עצמו.

	addi \$1 \$1 222																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			</
--	------------------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	----

נשים לב שהתוכן זהה מלבד רגיסטרים 28/29 ש-MARS משתמש למטרות אחרות ומלבד רגיסטר 31 ששמר את הכתובה חזרה לאחר פקודת JAL. כלומר התוצאה נכונה.

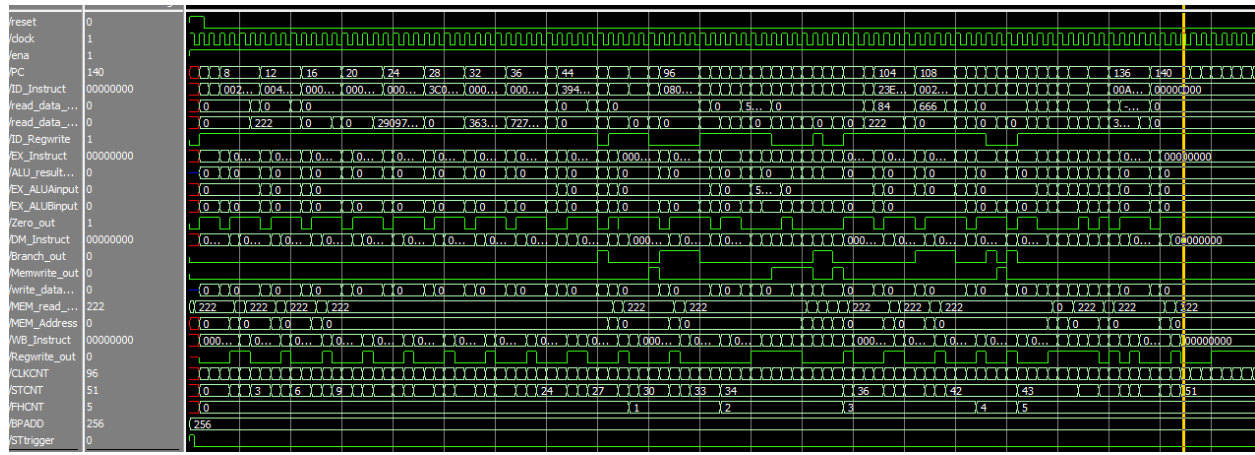
7	0
6	0
5	0
4	3637248
3	-536870912
2	222
1	444
	222

נביט בזיכרון בשתי המצבים, נראה שהוא גם זהה.

Data Segment					
Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)
0	222	444	222	-536870912	3637248
32	0	0	0	0	0
64	0	0	0	0	0

WAVE FORM

נביט בגדלים של הריצה.



רואים שכל פעם שיש לנו קפיצה ב-BRANCH אז המונה FHCNT גדל ב-1 כמצופה.
 רואים מהקוד, שכמעט כל פקודה מבצעת כתיבה לרגיסטרים ורואים זאת מהגלים, האות REGWRITE דולק כמעט תמיד.

אם להביט רק באותות שונים של INSTRUCTION ניתן לראות את הצורה האלכסונית של ביצוע הפקודה לפי כל רכיב. כמו כן גם סטולים/FLUSH שמתקיימים במהלך.

הערה: בקוד המקורה היו לנו את כל ה-5 שלבים אבל בקוד שמגישים התבקש רק 3 שלבים

00430820	23E10000	00000000	000422C0	014B682A	294E03E7	294F03E7	AC010000	AC020004	AC030008	AC04000C	0C000C18
	08000C0C	00000000			000422C0	014B682A	294E03E7	294F03E7	AC010000	AC020004	AC030008
		08000C0C	00000000			000422C0	014B682A	294E03E7	294F03E7	AC010000	AC020004

נביט ברגיסטרים שהוספנו כדי לחשב את ICP:

CLKCNT = 96

STCNT = 51

FHCNT = 5

$$IPC = \frac{CLKCNT - (STCNT + 4 + FHCNT * dept)}{CLKCNT} = \frac{96 - (51 + 4 + 5 * 3)}{96} = \frac{13}{48} \approx 0.271$$

התוצאה ד"י נמוכה, זאת משום שהקוד נכתב בכוונה עם הרבה תלות בין פקודות על מנת לסבך וליצור הרבה סטולים ולעשות קפיצות בתהליך.

דוגמה לקוד שלנו – diviation QA

הקוד מבצע את החישוב $16/2=8$ ללא פעולת div כמובן. הכנסנו עוד כמה פעולות מיותרים על מנת ליצור סטולים

מכוונים.

```
.data
result: .word 0      # Memory location to store the result

.text

addi $t0,$0, 16      # Load dividend value into $t0
addi $t1,$0, 2        # Load divisor value into $t1
addi $ra , $0,100

# Initialize variables
move $t2, $zero      # Quotient: $t2 = 0
move $t3, $t0        # Remainder: $t3 = $t0
addi $t7, $0,1

loop:
    bne $t3, $t1, subtract # Branch to subtract if remainder
    addi $zero,$zero,0
    addi $t2, $t2, 1      # Increment quotient
    j end                # Jump to end

subtract:
    sub $t3, $t3, $t1    # Subtract divisor from remainder
    addi $t2, $t2, 1      # Increment quotient
    sub $t2 , $t2,$t7     #stall check
    addi $t2,$t2,1
    j loop              # Jump to loop

end:
    addi $zero,$zero,0
    add $t3, $t3, $t1     # Add divisor to remainder

    sw $t2, result       # Store the result in memory location
    jr $ra               # Return
```

Number	Value
0	0
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	16
9	2
10	8
11	2
12	0
13	0
14	0
15	1
16	0
17	0
18	0
19	0
20	0
21	0
22	0
23	0
24	0
25	0
26	0
27	0
28	6144
29	12284
30	0
31	100
	12364
	0
	0

Memory Data - /mipstb/L1/ID/register	
0	0
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	16
9	2
10	8
11	4
12	0
13	0
14	0
15	1
16	0
17	0
18	0
19	0
20	0
21	0
22	0
23	0
24	0
25	0
26	0
27	0
28	0
29	0
30	0
31	100

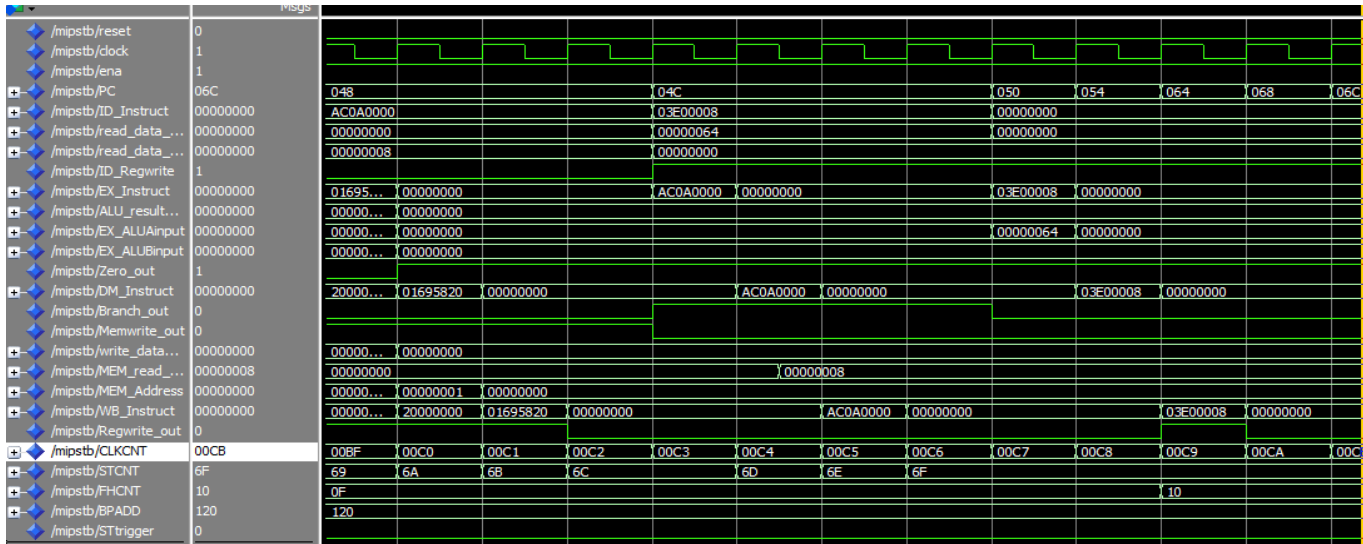
תוכן הרגיסטרים והזיכרון נתונים זהה.

Address	Value (+0)
0	8

1 | 8

WAVE FORM

נביט בגדלים של הריצה.



$$IPC = \frac{CLKCNT - (STCNT + 4 + FHCNT * dept)}{CLKCNT} = \frac{203 - (111 + 4 + 10 * 3)}{203} = 0.28$$

קיבלנו מספר קטן יחסית כי בכל לולאה ישנם 2 פקודות קפיצה היוצרות סטול ופלאש
ובנוסף הכנסנו פקודת קריאה-כתיבה היוצרות סטול נוסף

/mipstb/CLKCNT	00CB
/mipstb/STCNT	6F
/mipstb/FHCNT	10