



פרויקט סיום

MIPS based MCU

Architecture and Design

שמות המגישים והעורכים:

נועם פרץ, נעם וקנין

המחלקה להנדסת חשמל – קורס מעבדת ארכיטקטורת מעבדים מתקדמת
ומאיצי חומרה

תאריך הגשת הדו"ח: 08.08.23

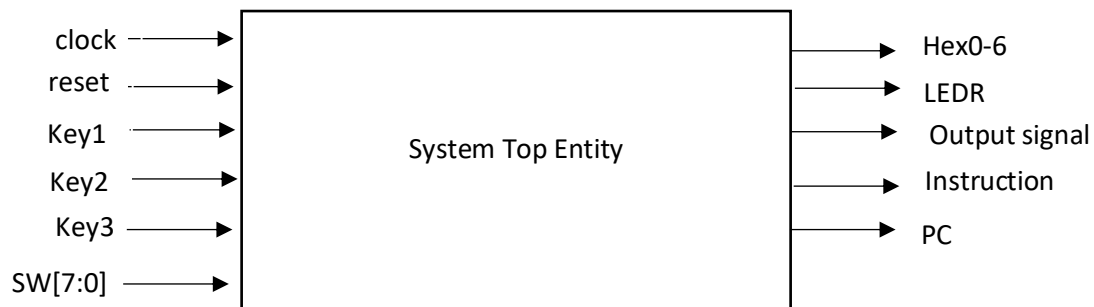


תוכן עניינים

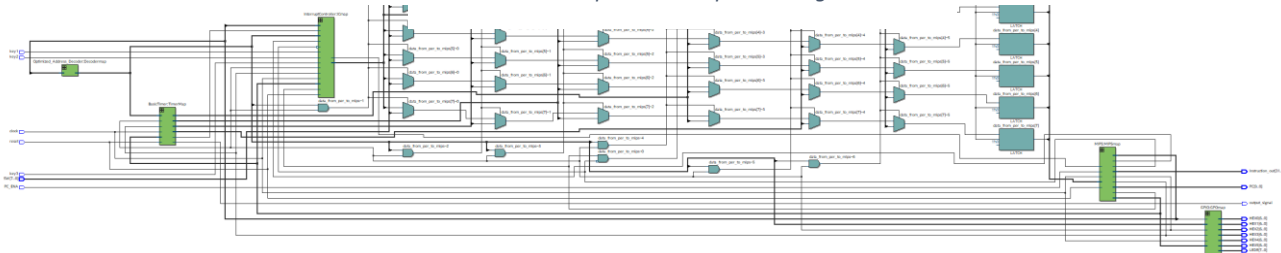
3.....	MCU Top level block	.1
4.....	Blocks in top level design	.2
15.....	Critical Path and Fmax	.3
16.....	Waveform ModelSim	.4
18.....	Waveform Quartus Signal Tap	.5



MCU block diagram



MCU Graphical description :1 Figure



MCU RTL :2 Figure

Flow Summary	
<<Filter>>	
Flow Status	Successful - Tue Aug 8 05:06:12 2023
Quartus Prime Version	21.1.0 Build 842 10/21/2021 SJ Lite Edition
Revision Name	FPGA_interface
Top-level Entity Name	MCU
Family	Cyclone V
Device	5CSXFC6D6F31C6
Timing Models	Final
Logic utilization (in ALMs)	1,446 / 41,910 (3 %)
Total registers	1173
Total pins	107 / 499 (21 %)
Total virtual pins	0
Total block memory bits	98,304 / 5,662,720 (2 %)
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 %)

MCU Logic Usage :3 Figure



MIPS block diagram

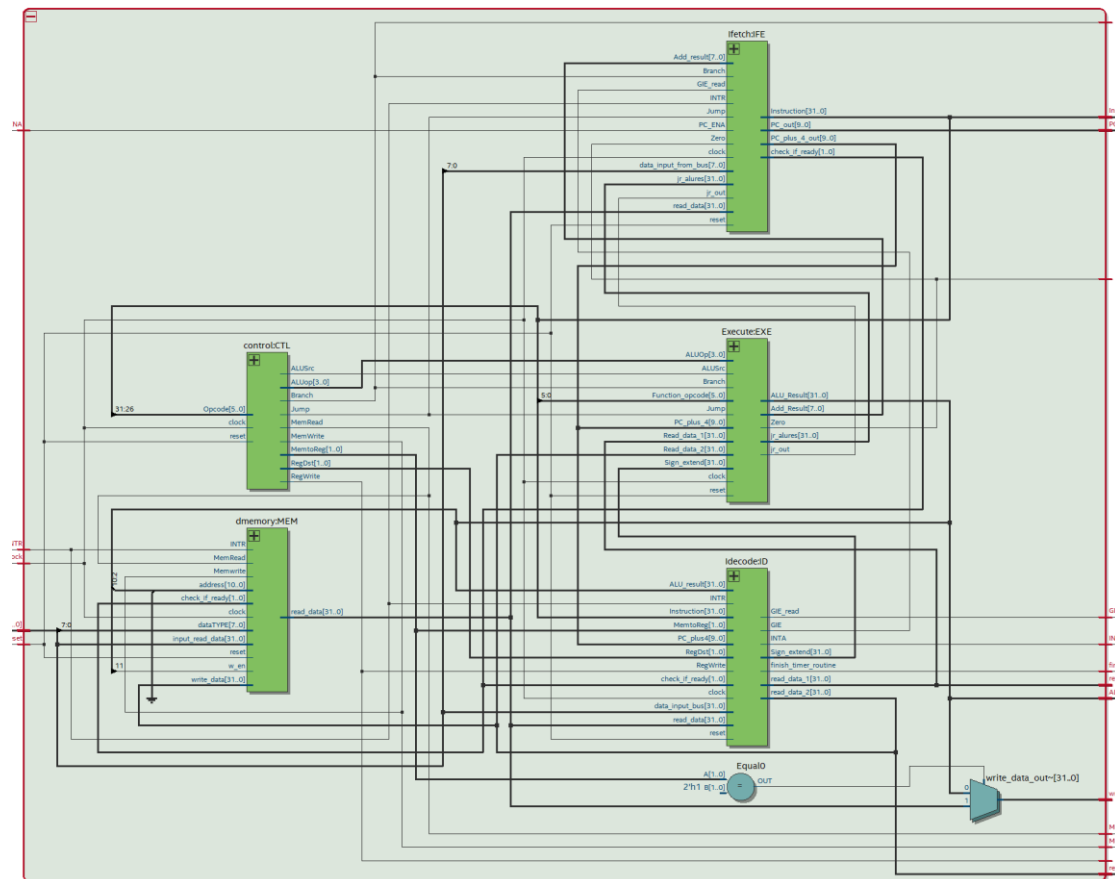


Figure 4 :MIPS RTL

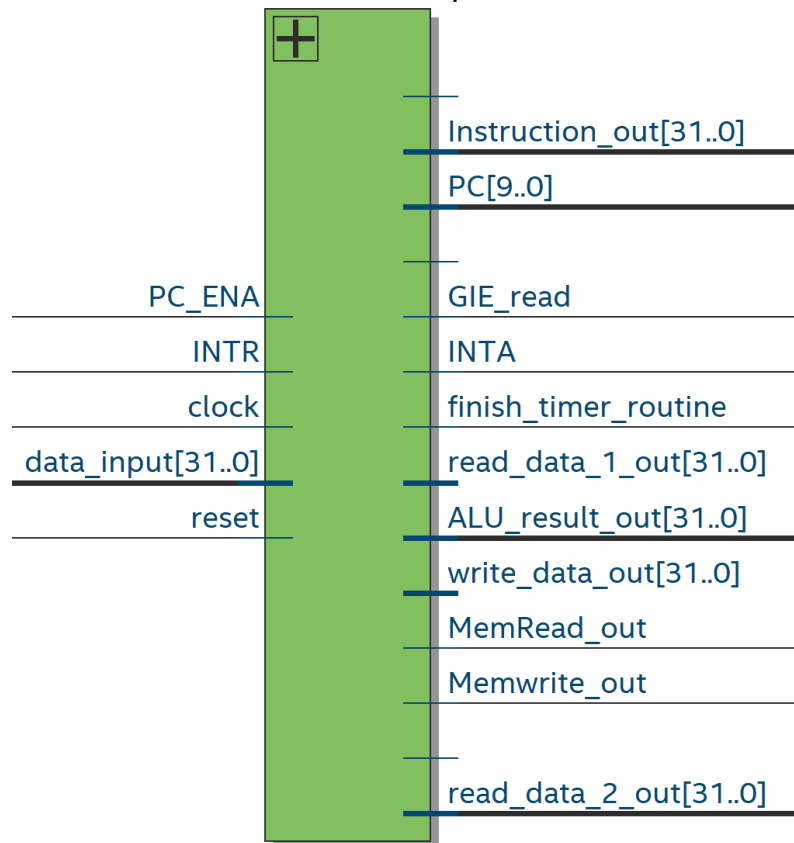
נבחין ברכיבים של הIftech, Execute, Idecode, Dmemory, Control כפי שראינו במעבדה.
5.

	Compilation Hierarchy Node	Combinational ALUTs	Dedicated Logic Registers	Block Memory Bits	SP Block
1	▼ MCU	1931 (83)	1169 (0)	98304	2
1	▶ BasicTimer:TimerMap	216 (213)	109 (106)	0	0
2	▶ GPIO:GPOmap	52 (10)	32 (32)	0	0
3	▶ InterruptController:ICmap	42 (42)	25 (25)	0	0
4	▶ MIPS:MIPSmapi	1533 (0)	1003 (0)	98304	2

Figure 5 :MIPS Logic Usage



MIPS:MIPSmip



MIPS Graphical description :6 Figure

מודול זה זהו הליבה של המעבד (CPU Core), אחראית לרכיבים שתיארנו למעלה תוך גישה ל-2 מרחבי זיכרון, של הפקודות (program mem) ושל המידע (data mem). מוציאה החוצה את ערך הנתונים הנוכחי בעת כתיבה לרכיבים הפריפריאליים.

יחד עם זאת כל הרכיבים מאזינים לכתובת שיוצאת מרכיב ולאותות הבקרה של כתיבה וקריאה.

:Port Table

Port Name	Direction	Size	Functionality
PC	OUT	10 bit	Program Counter output
ALU_result_out	OUT	32 bit	ALU output
Read_data_1_out	OUT	32 bit	Output read data 1
Read_data_2_out	OUT	32 bit	Output read data 2
Write_data_out	OUT	32 bit	Output of data to be written
Instruction_out	OUT	32 bit	Output of instruction
reset	IN	bit	reset
clock	IN	bit	clock
Branch_out	OUT	bit	Output indicating branch condition
Zero_out	OUT	bit	Zero condition

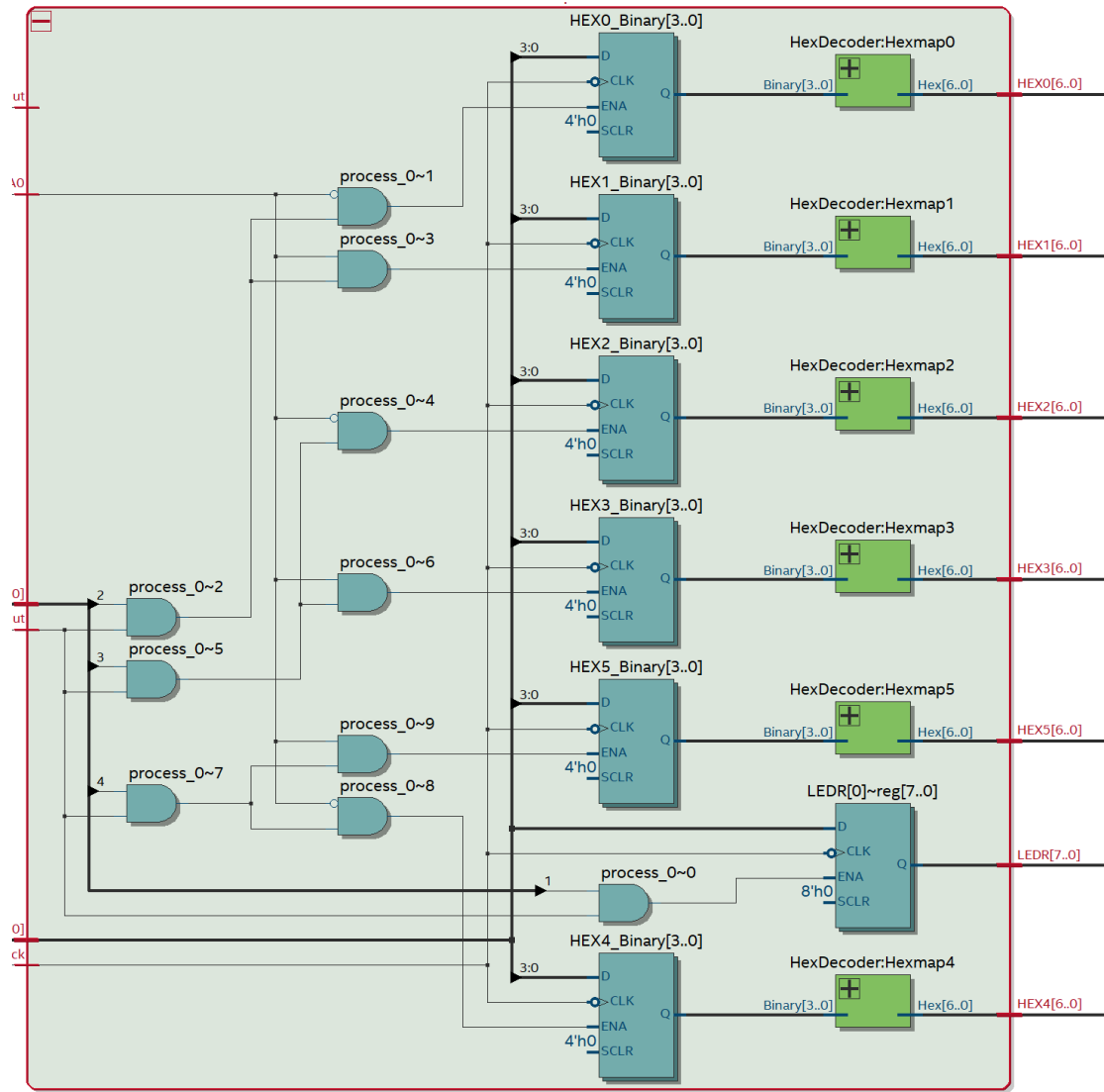


Memwrite_out	OUT	bit	Memory write operation
Regwrite_out	OUT	bit	Register write operation
PC_ENA	IN	bit	Enable signal for PC
MemRead_out	OUT	bit	Indicating memory read operation
Data_input	IN	32 bits	Input data from the bus
GIE_read	OUT	Bit	Output indicating GIE status
INTR	IN	Bit	Interrupt request input
INTA	OUT	1 bit	Interrupt acknowledge output
Finish_timer_routine	OUT	bit	Output indicating completion of timer routine

MIPS Port Table :Figure 7



GPIO block diagram



GPIO RTL :8 Figure

Compilation Hierarchy Node	Combinational ALUTs	Dedicated Logic Registers	Block Memory Bits
1 MCU	1931 (83)	1169 (0)	98304
1 BasicTimer:TimerMap	216 (213)	109 (106)	0
2 GPIO:GPOMap	52 (10)	32 (32)	0

Figure 9 :GPIO Logic Usage

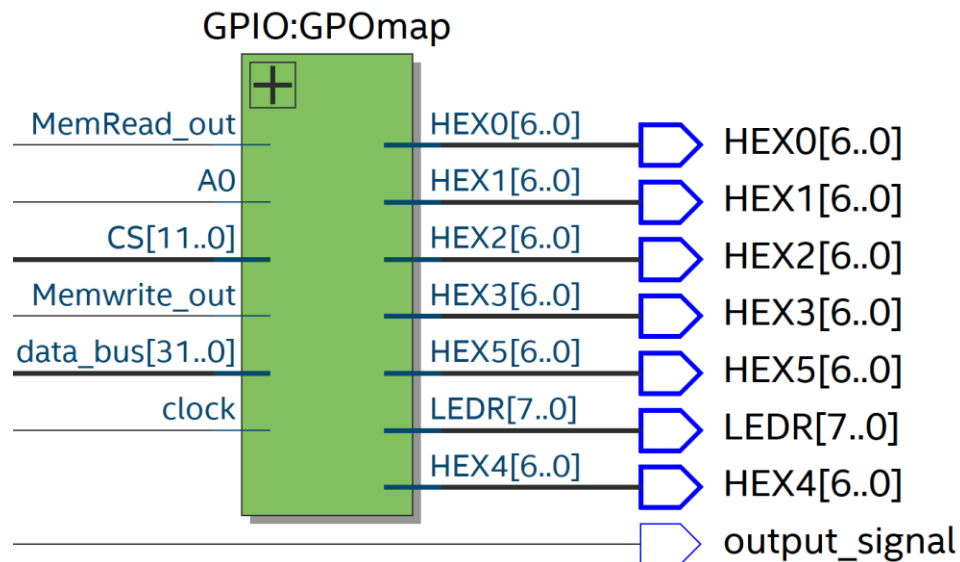


Figure 10 :GPIO Graphical description

בעזרת רכיב זה נכתוב להקסות (בעזרת HexDecoder) וללדים ע"י צ'יפ סלקט כפי שתואר במטלה.

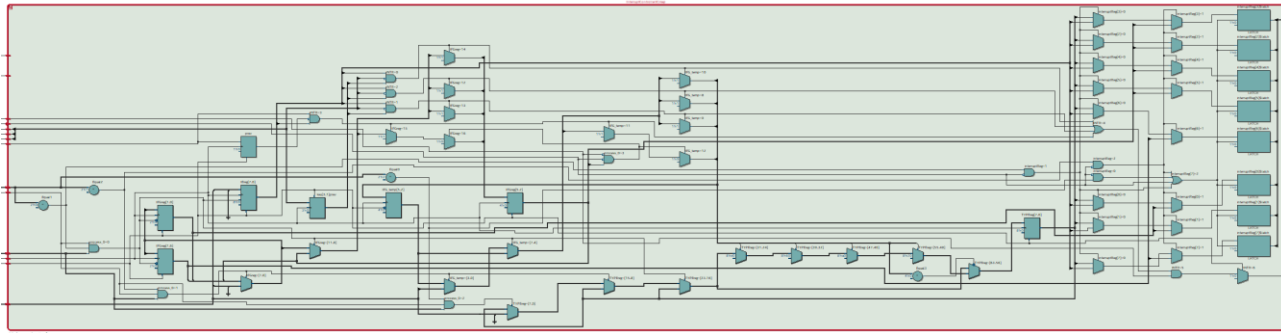
:Port Table

Port Name	Direction	Size	Functionality
CS	IN	12 bit	Chip select input
A0	IN	Bit	Address bit 0 input
MemRead_out	IN	Bit	Memory read signal input
Memwrite_out	IN	Bit	Memory write signal input
Data_bus	IN	32 bit	Data bus input
LEDR	OUT	8 bit	LED outputs for display
HEX0-HEX5	OUT	7 bits each	Seven-segment display outputs
Clock	IN	Bit	Clock

GPIO Port Table :Figure 11



Interrupt Controller block diagram

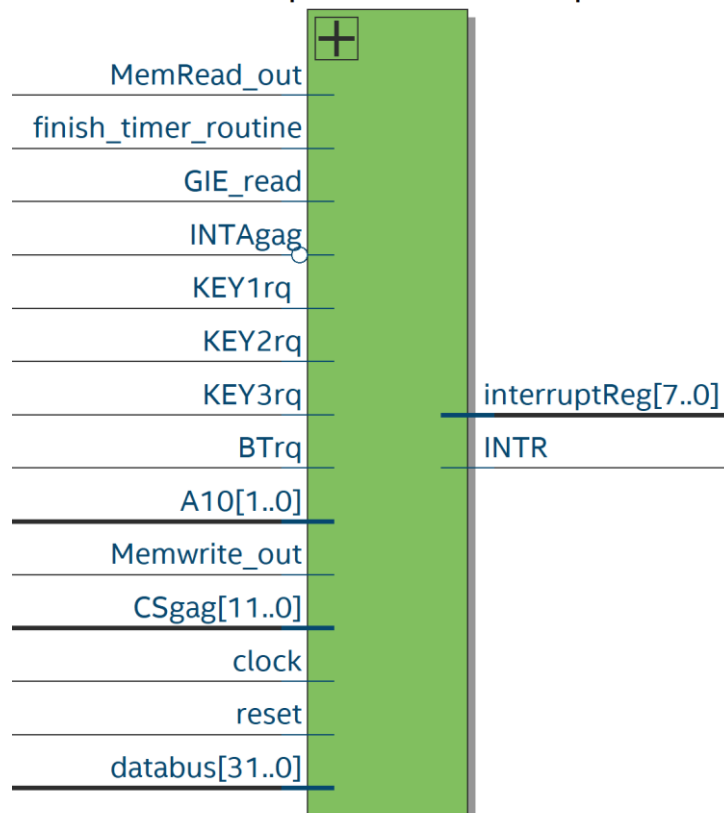


Interrupt Controller RTL :12 Figure

	Compilation Hierarchy Node	Combinational ALUTs	Dedicated Logic Registers	Block Memory Bits	SP Block
1	▼ MCU	1931 (83)	1169 (0)	98304	2
1	▶ BasicTimer:TimerMap	216 (213)	109 (106)	0	0
2	▶ GPIO:GPOmap	52 (10)	32 (32)	0	0
3	▶ InterruptController:ICmap	42 (42)	25 (25)	0	0
4	▶ MISC:MISCmap	1522 (10)	1002 (0)	65536	2

Interrupt Controller Logic Usage :13 Figure

InterruptController:ICmap



Interrupt Controller Graphical description :14 Figure



רכיב זה אחראי למנגנון הפסיקות, הוצאת הביט INTR ל CPU בעת פסיקה וקבלת \overline{INTA} כאשר הפסיקה התקבלה. פסיקה יכולה להיות מהטיימר או מהKeys. בנוסף הוא אחראי לדאוג לעדיפות הפסיקות.

:Port Table

Port Name	Direction	Size	Functionality
Clock	IN	Bit	Clock input
Reset	IN	Bit	Reset input
CSgag	IN	12 bit	Complement of chip select input
BTrq	IN	Bit	BasicTimer interrupt request input
Key1rq	IN	Bit	KEY1 interrupt request input
Key2rq	IN	Bit	KEY2 interrupt request input
Key3rq	IN	Bit	KEY3 interrupt request input
INTAgag	IN	Bit	Complement of interrupt acknowledge input
INTR	OUT	Bit	Interrupt request output
Databus	IN	32 bit	Data bus input
A10	IN	2 bit	Address bits A1-A0 input
GIE_read	IN	Bit	GIE read input
interruptReg	OUT	8 bit	Interrupt flag, interrupt enable, type output
MemRead_out	IN	Bit	Memory read input
Memwrite_out	IN	Bit	Memory write input
Finish_timer_routine	IN	bit	Input indicating completion of timer routine

Interrupt Controller Port Table :Figure 15



Basic Timer block diagram

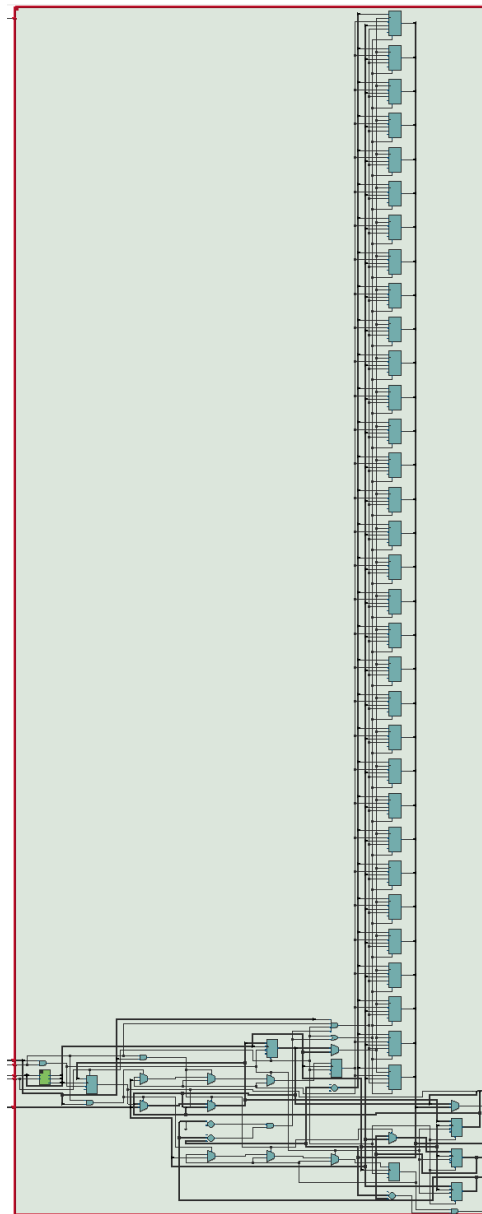


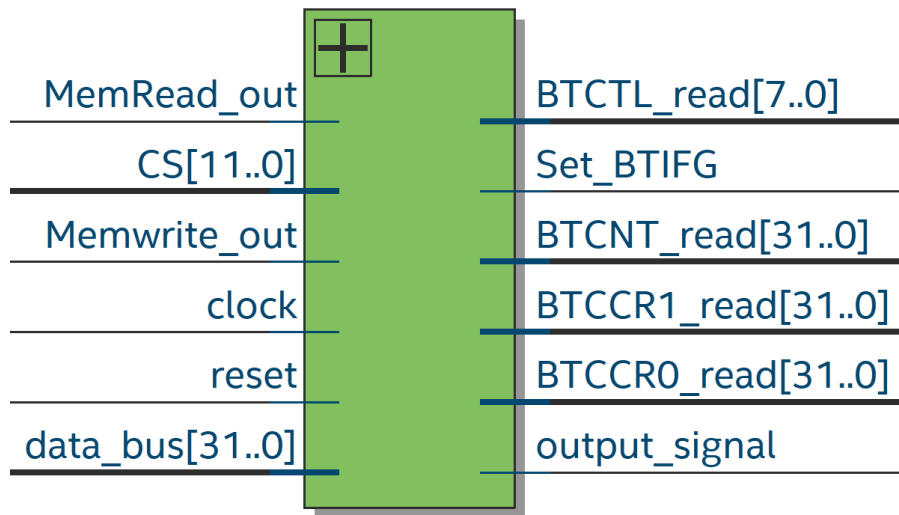
Figure 16 :Basic Timer RTL

	Compilation Hierarchy Node	Combinational ALUTs	Dedicated Logic Registers	B
1	▼ MCU	1931 (83)	1169 (0)	98
1	► BasicTimer:TimerMap	216 (213)	109 (106)	0
2	► GPIO:GPIOmap	52 (10)	22 (22)	0

BasicTimer Logic usage :17 Figure



BasicTimer:TimerMap



Basic Timer Graphical description :18 Figure

רכיב זה אחראי להוציא פסיקות ע"י הביט Set_BTIFG ולהוציא אות PWM לפי מנייה עולה כפי שתואר במטלה.

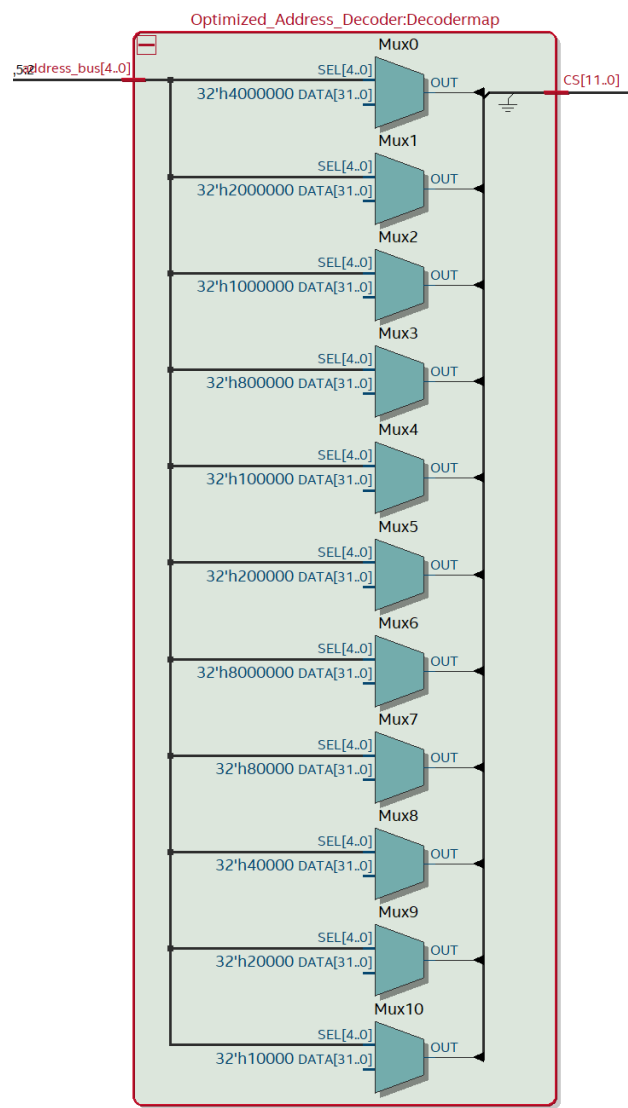
:Port Table

Port Name	Direction	Size	Functionality
Clock	IN	Bit	Clock input
Reset	IN	Bit	Reset input
CS	IN	12 bit	chip select input
BTCTL_read	OUT	8 bit	BT control register read output
BTCCR0_read	OUT	32 bit	BT compare register 0 read output
BTCCR1_read	OUT	32 bit	BT compare register 1 read output
BTCNT_read	OUT	32 bit	Current count value read output
MemRead_out	IN	Bit	Memory read input
Memwrite_out	IN	Bit	Memory write input
Databus	IN	32 bit	Data bus input
Output_signal	OUT	Bit	Output signal based on duty cycle
Set_BTIFG	OUT	bit	Output to set BTIFG interrupt flag

BasicTimer Port Table :Figure 19



Optimized Address Decoder block diagram



Optimized Address Decoder RTL :20 Figure

Compilation Hierarchy Node	Combinational ALUTs	
▼ MCU	1931 (83)	1'
▶ BasicTimer:TimerMap	216 (213)	10
▶ GPIO:GPOmap	52 (10)	32
InterruptController:ICmap	42 (42)	25
▶ MIPS:MIPSmap	1533 (0)	10
Optimized_Ad...r:Decodermap	5 (5)	0

Optimized Address Decoder Logic usage :21 Figure



Optimized_Address_Decoder:Decodermap

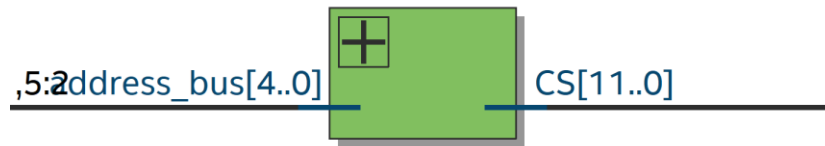


Figure 22 :Optimized Address Decoder Graphical description

רכיב זה מקבל כתובת ומקודד אותו לצ'יפ סלקט.

:Port Table

Port Name	Direction	Size	Functionality
Address_bus	IN	5 bits	Address bus input
CS	OUT	12 bits	CS outputs

Optimized Address Decoder :Figure 23



Critical Path and f_{max}

Slow 1100mV 85C Model Fmax Summary

<<Filter>>

	Fmax	Restricted Fmax	Clock Name	Note
1	24.6 MHz	24.6 MHz	clock	

Maximum clock frequency :24 Figure

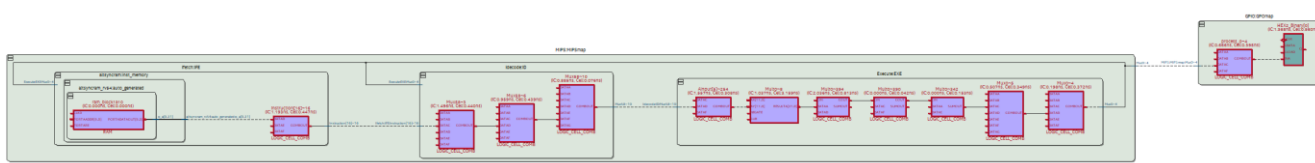
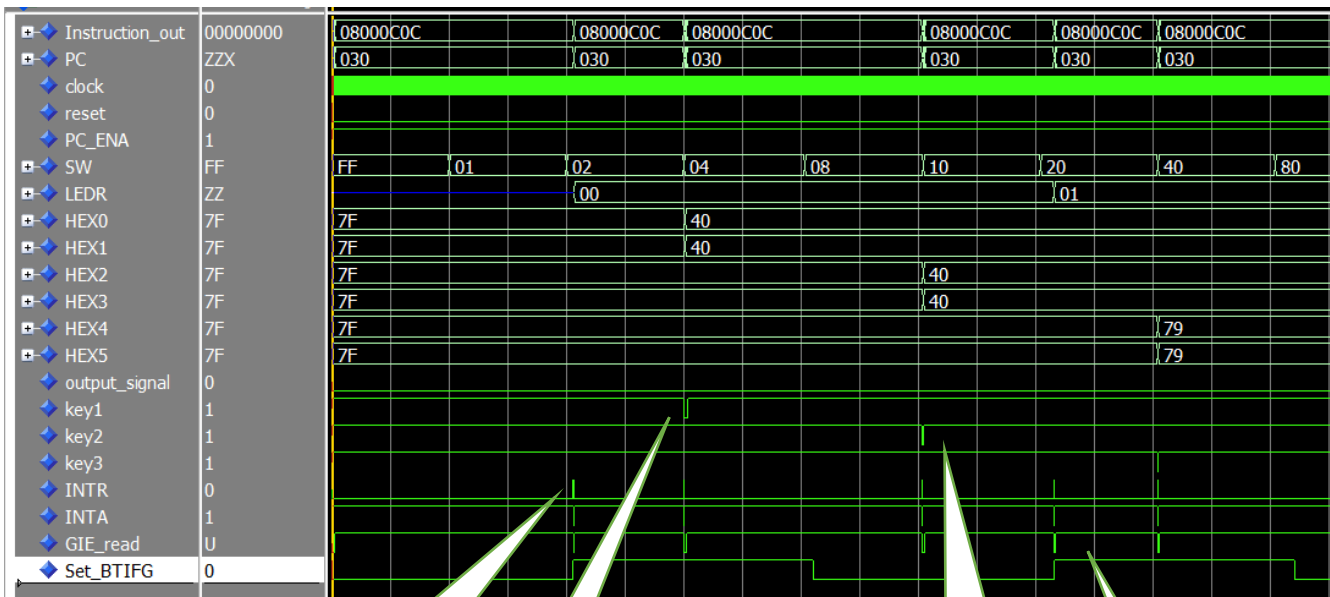


Figure 25 :MCU Critical Path

נשים לב שהמסלול הקריטי הוא במעבר בכל התחנות בMIPS ושליחת המידע למודול של GPIO (לכתיבה), קיבלנו תוצאה הגיונית.



Waveform ModelSim



ModelSim Waveform :26 Figure

פסיקה מהטיימר
וסיומה כאשר ערך
הלדים מתעדכן ל00

פסיקה מkey1 וסיומה
כאשר ערך הקס0,1
מתעדכנים ל40

פסיקה מkey2 וסיומה
כאשר ערך הקס0,1
מתעדכנים ל40

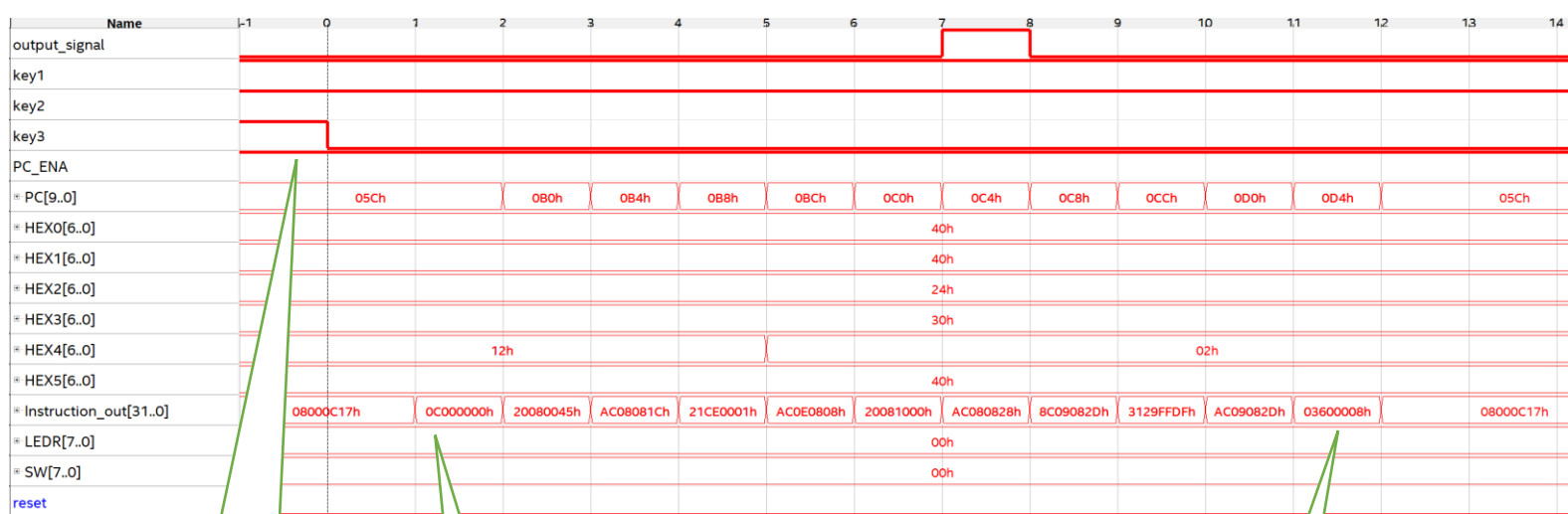
פסיקה נוספת
מהטיימר ובסיומה
פסיקה של key3



ניתן לראות שאכן לאחר כל פסיקה ה-PC חוזר לכתובת שממנה יצא לפסיקה. וכל פסיקה מתבצעת כראוי כך שפסיקה . נראה זאת גם ב-Signal Tap.



Waveform Quartus Signal Tap



Waveform Signal Tap :Figure 27

פסיקה מkey3
שבסיומה מתעדכן
הערך של Hex4

תחילת ביצוע הפסיקה,
ביצוע פקודת jal, שמירת
ערך הPC ברגיסטר 27

סיום ביצוע הפסיקה עם
הפקודה jr 27 שאכן
לאחריה חזרנו לPC
שממנו קפצנו

בנוסף נוכל להבחין בlatency של 2 מחזורי שעון מרגע לחיצת הכפתור לפסיקה ועד אחרי פקודה הjal. latency של מחזור אחד בחזרה בפקודת jr.

לסיכום, בעזרת פרויקט זה למדנו כיצד מחברים ומתפעלים רכיבים פריפריילים, IO, לליבת MIPS שבנינו. בנוסף התמקצענו במימוש פסיקות מרכיבים אלה בעזרת ה- Interrupt Controller וקווי הפסיקות INTR, INTA.