

המחלקה להנדסת חשמל

**Department of Electrical Engineering**

## **דוח פרויקט מעבדה בהנדסת חשמל**

**Final Lab project – Mouse Controller**

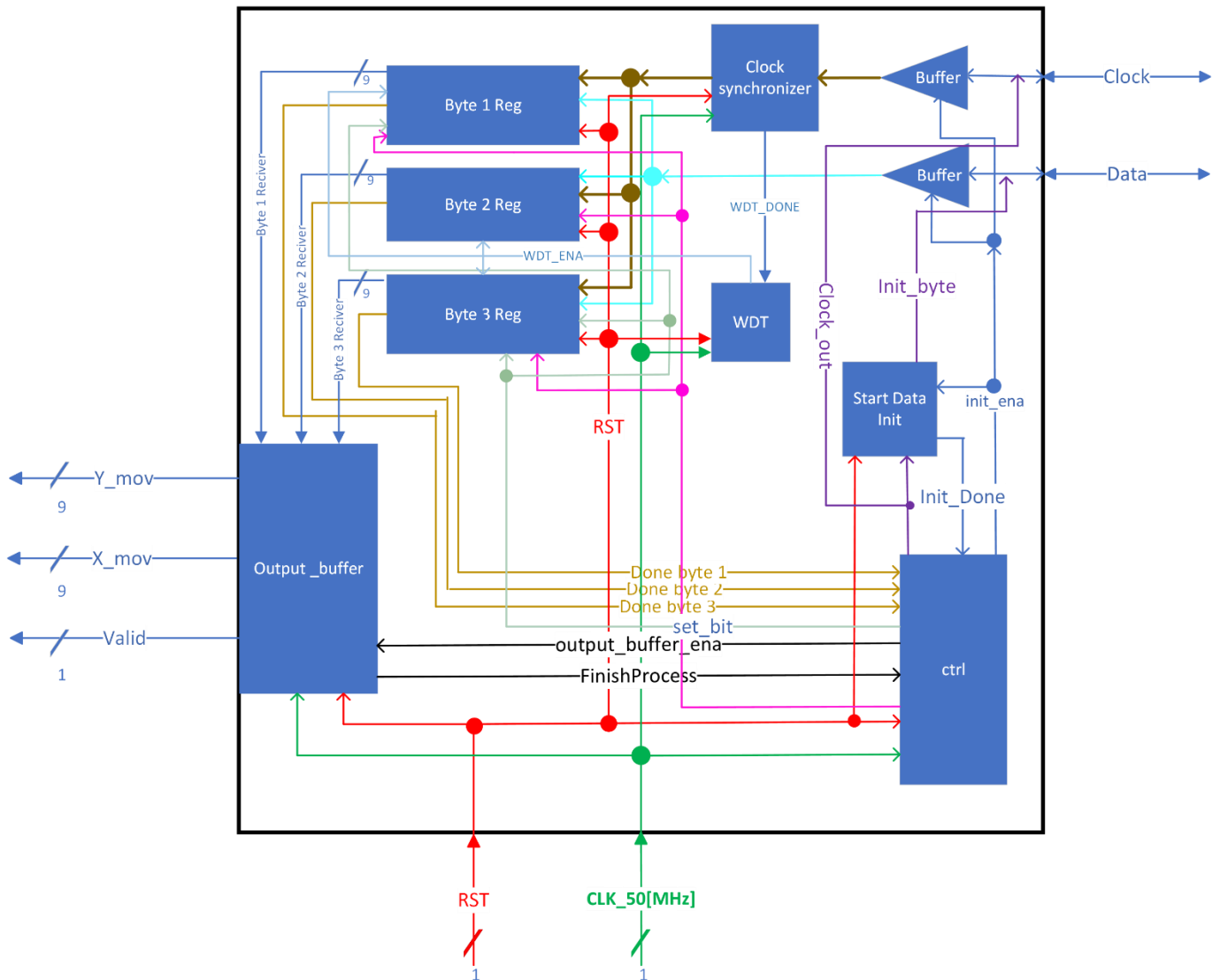


## תוכן עניינים:

3	.....	<a href="#">1. ארכיטקטורת של המערכת</a>
3	.....	<a href="#">1.1. שרטוט דיאגרמת בלוקים של המערכת</a>
4	.....	<a href="#">1.2. שרטוט RTL של המערכת</a>
5	.....	<a href="#">2.2. דיאגרמת מצבים</a>
5	.....	<a href="#">2.1. שרטוט דיאגרמת מצבים</a>
6	.....	<a href="#">2.2. שרטוט RTL של דיאגרמת מצבים</a>
7	.....	<a href="#">3. פעולת המערכת</a>
12	.....	<a href="#">4. רכיבי המערכת</a>
12	.....	<a href="#">4.1. Clock Synchronizer</a>
12	.....	<a href="#">4.2. Watchdog Timer (WDT)</a>
13	.....	<a href="#">4.3. SIPO(Serial-In Parallel-Out) Registers</a>
13	.....	<a href="#">4.4. Output Buffer</a>
14	.....	<a href="#">4.5. Init</a>
14	.....	<a href="#">4.6. Main ctrl unit</a>

# 1. ארכיטקטורת של המערכת

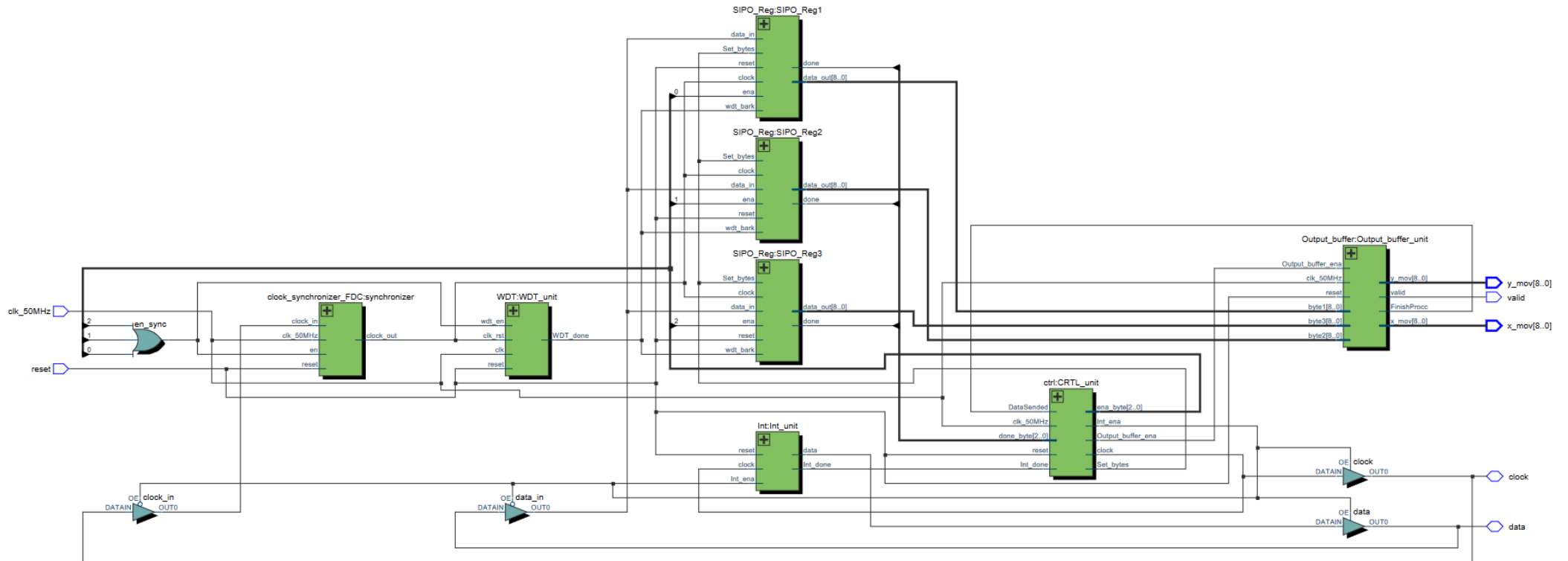
## 1.1. שרטוט דיאגרמת בלוקים של המערכת



ליחידה קיימים אותות in out של clock & data כניסות ,clk\_50[MHz] ,reset ו מוצאים X\_mov, Y\_mov, Valid. יחידת CTRL שהיא מערכת המצבים שלנו, יחידת אתחול המערכת ושליחת הודעה לחיבור העכבר עם המחשב, יחידת סנכרון שעון, יחידת בטיחות לעבודה סיסטמתית של המערכת WDT, רגיסטרים מסוג SIPO, ויחידת האחראית למוצא הסיגנלים של תזוזת העכבר יחד עם אות המייצג שניתן לקרוא את המוצא התקין.



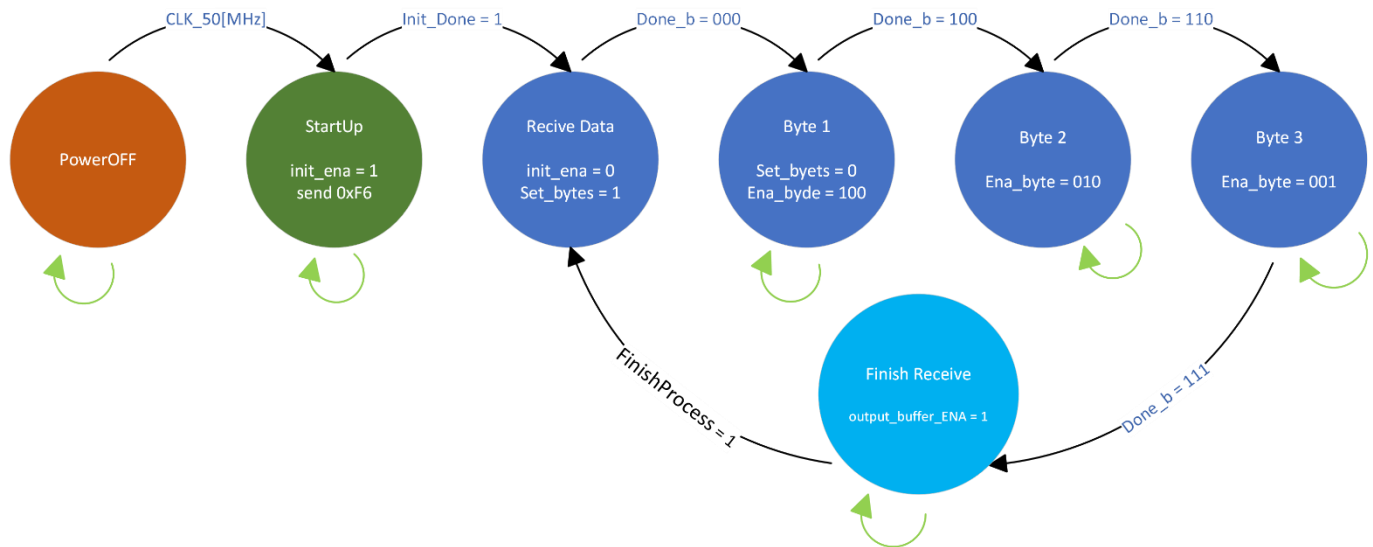
## 1.2. שרטוט RTL של המערכת



## 2. דיאגרמת מצבים

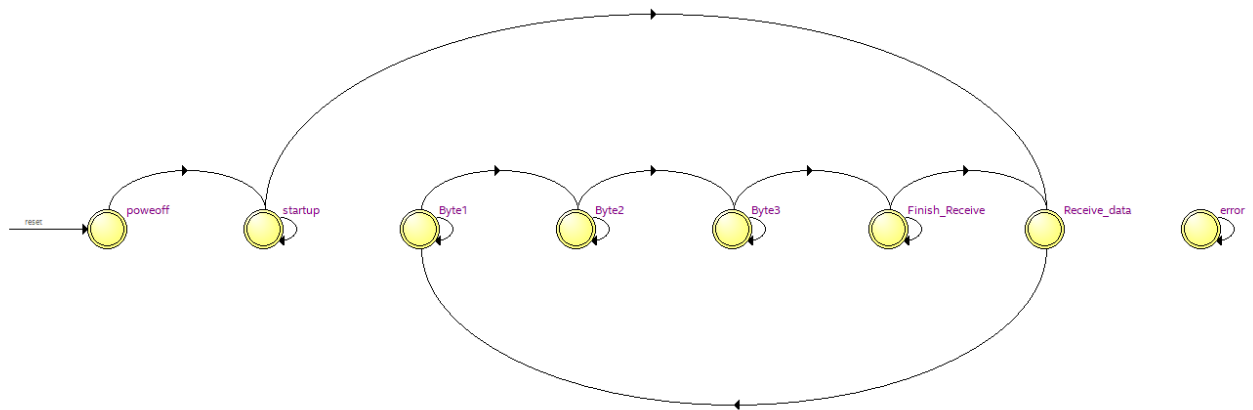
### 2.1. שרטוט דיאגרמת המצבים

#### Mealy machine



Current State	Next State	State Move Condition
Power Off	Power Off	No Init trigger
Power Off	Startup	Init Triger
Startup	Startup	Init_done = '0'
Startup	Receive Data	Init_done = '1'
Receive Data	Byte 1	Clock pulse
Byte 1	Byte 1	(!done_b[0])
Byte 1	Byte 2	done_b[0]
Byte 2	Byte 2	(!done_b[1])
Byte 2	Byte 3	done_b[1]
Byte 3	Byte 3	(!done_b[2])
Byte 3	Finish Receive	done_b[2]
Finish Receive	Finish Receive	(!FinishProcess)
Finish Receive	Receive Data	FinishProcess

## 2.2. שרטוט RTL של דיאגרמת המצבים

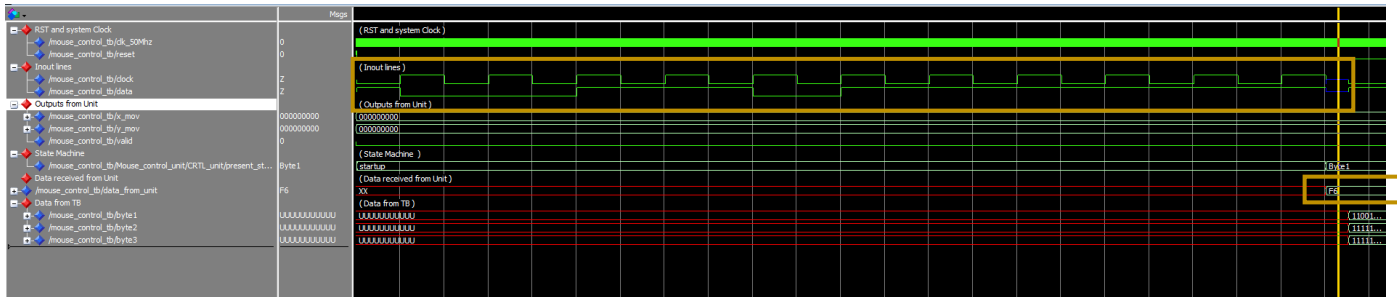


	Source State	Destination State	Condition
1	Byte1	Byte2	(done_byte[0])
2	Byte1	Byte1	(!done_byte[0])
3	Byte2	Byte3	(done_byte[1])
4	Byte2	Byte2	(!done_byte[1])
5	Byte3	Finish_Receive	(done_byte[2])
6	Byte3	Byte3	(!done_byte[2])
7	error	error	
8	Finish_Rec...	Finish_Receive	(!DataSended)
9	Finish_Rec...	Receive_data	(DataSended)
10	poweoff	startup	
11	Receive_data	Byte1	
12	startup	startup	(!statmachine)
13	startup	Receive_data	(statmachine)

### 3. פעולת הרכיב

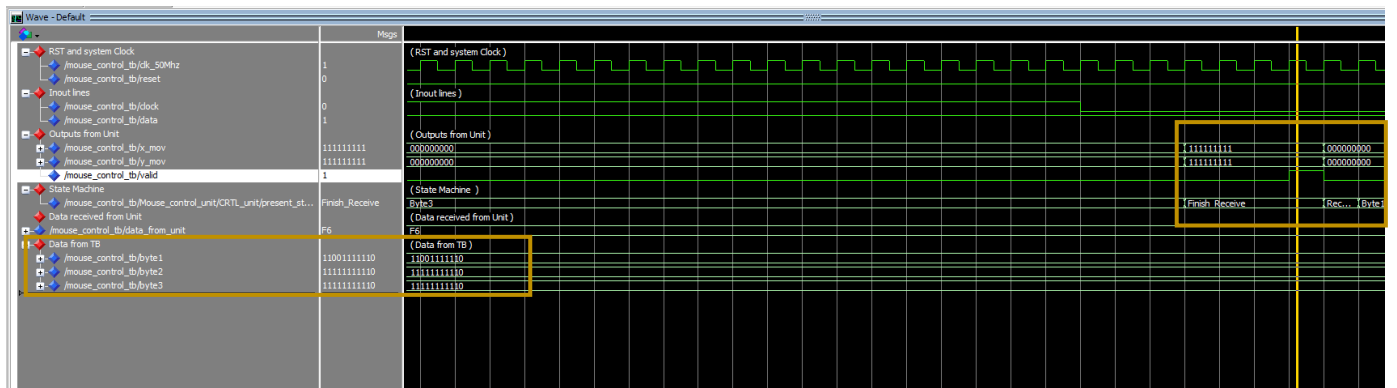
End Data Read when Data on '1' Parity bit Start Data Read when Data go from '1' to '0'

Startup & sending 0xF6(10111101100) Init signal:



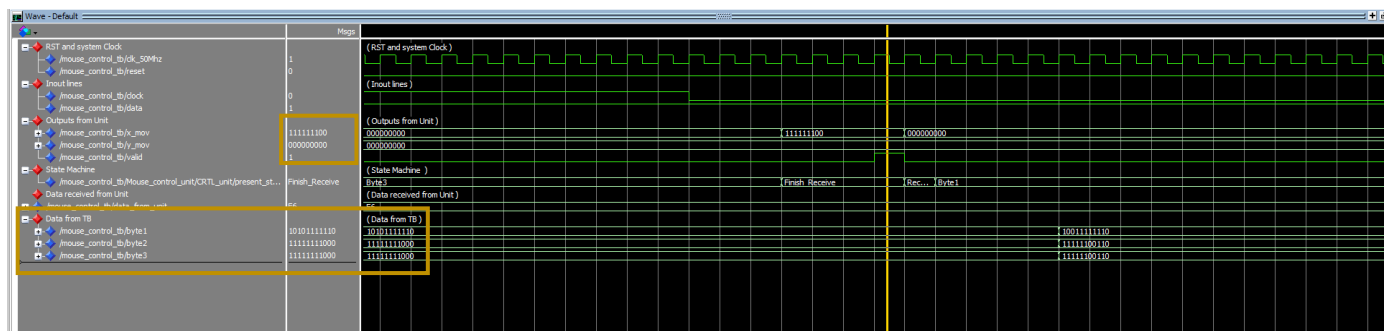
At first the unit is sending an Init signal of 0xF6 to tell the computer that the mouse is connected

#### Case 1:



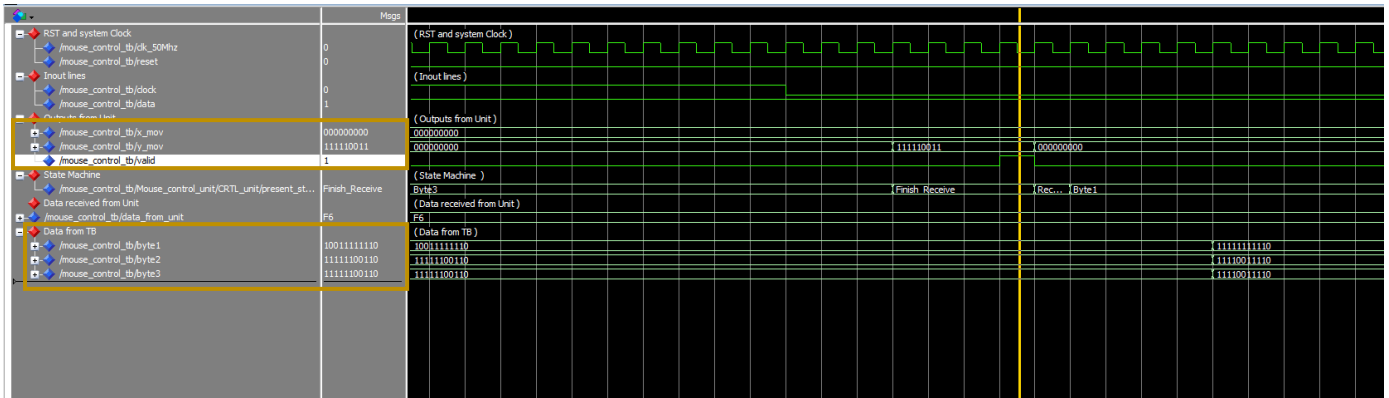
In this case all input data are valid, and no overflow bits are active. So, at the end of the state machine cycle we can see a Valid signal that is active for one clock pulse, x\_mov & y\_mov data as it receives it from outside.

#### Case 2:



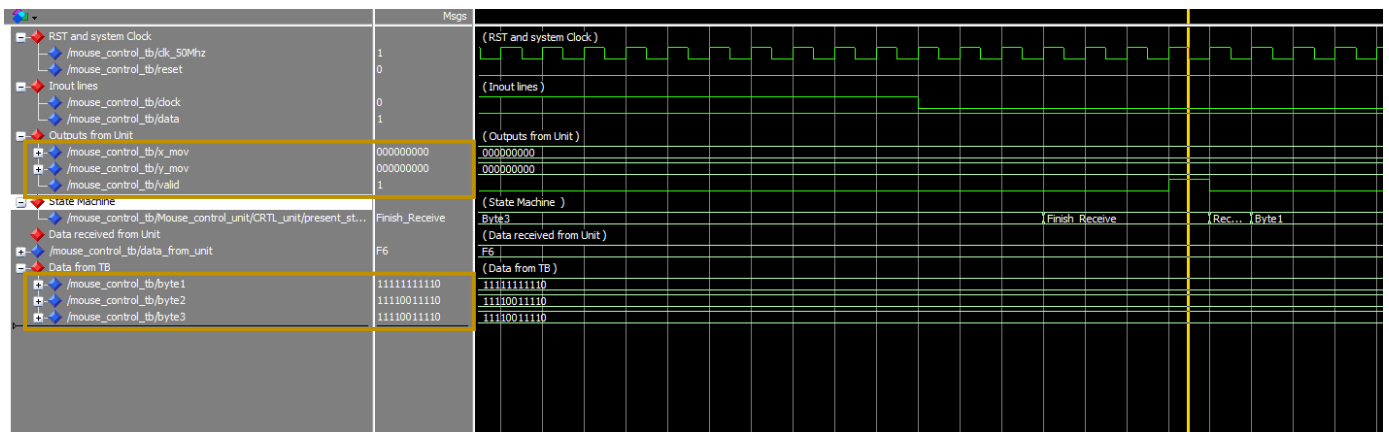
In this case all input data are valid and the y\_overflow bit is '1'. Valid signal is active for one cycle, x\_mov data is as it receives it from outside but y\_mov data is all 0.

### Case 3:



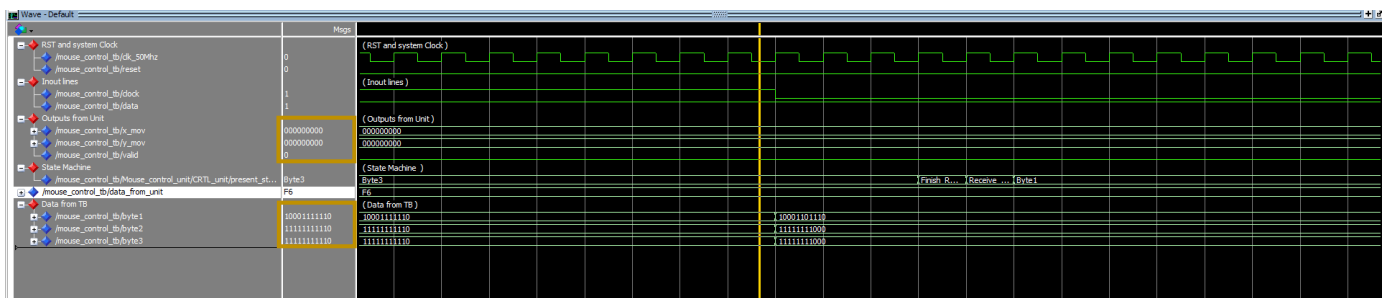
In this case all input data are valid and the x\_overflow bit is '1'. Valid signal is active for one cycle, y\_mov data is as it receives it from outside but x\_mov data is all 0.

### Case 4:



In this case all input data are valid and both x\_overflow & y\_overflow bits are '1'. Valid signal is active for one cycle, y\_mov & x\_mov data are all 0.

### Case 5:

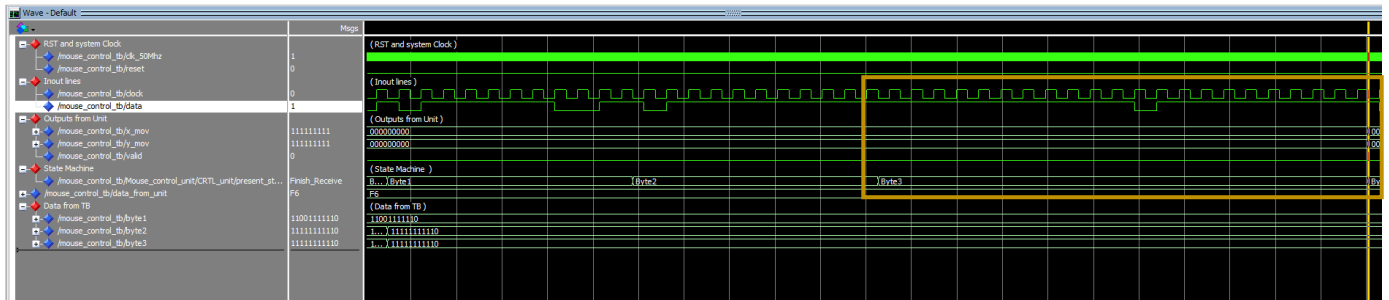


In this case byte1 parity bit is opposite to what is needed it to be and no active overflow bits. As you can see the valid signal is inactive and the outputs x\_mov & y\_mov are all '0'.



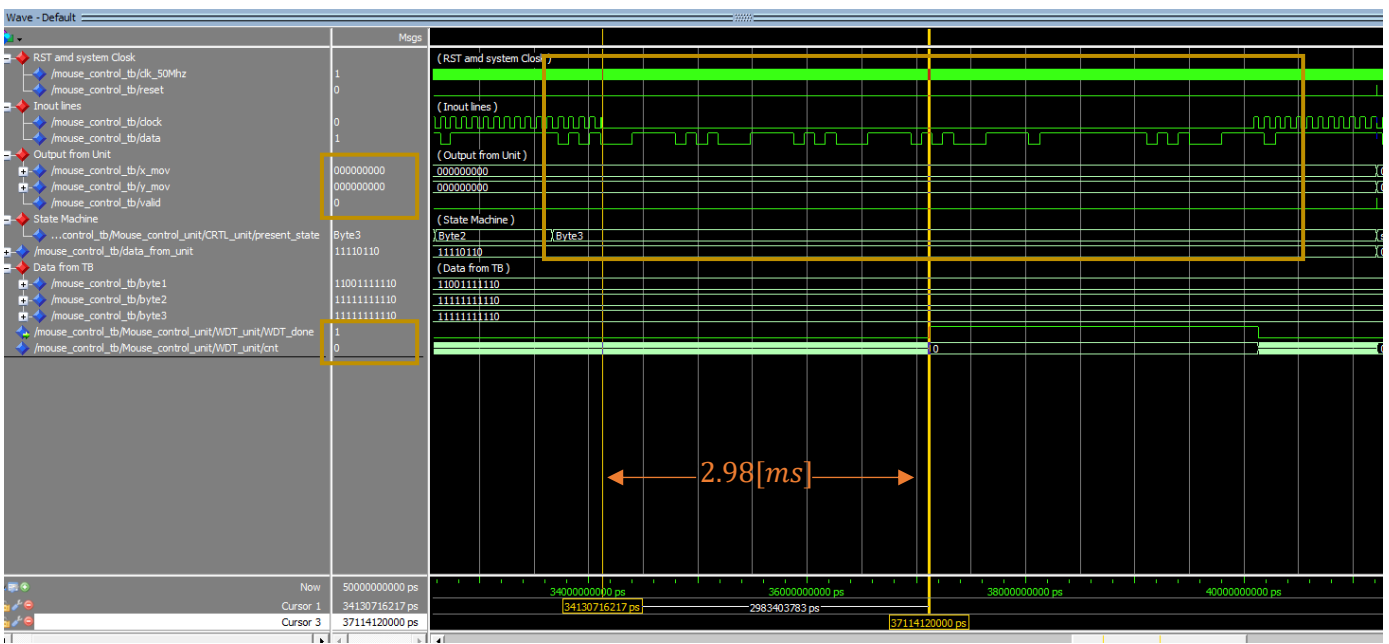


## Case 9:

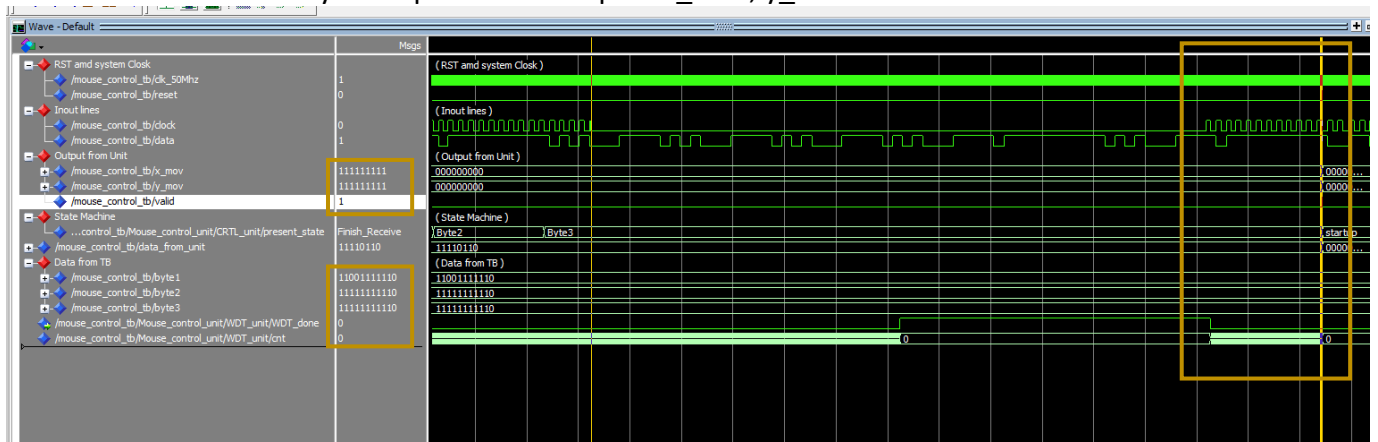


In this case we see that only when the data goes from `1` to `0` the read cycle of the byte starts (byte3). Otherwise.

## Case 10 (WDT Check):

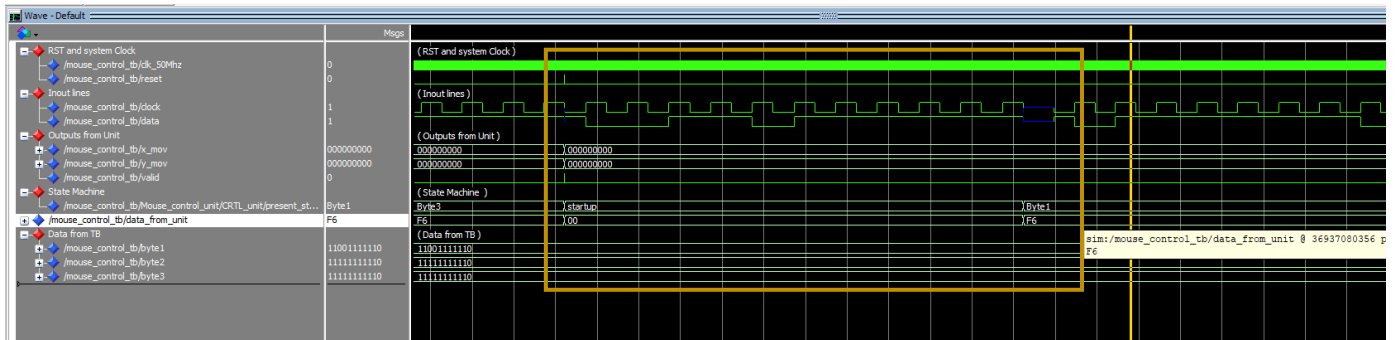


In this case we see the effect of WDT unit if the clock pulse in stays at `0` for too long (150000 cycles), the WDT signal is go up to `1` until it sees that the clock start running again then he starts to work properly as before the error for every clock pulse. The outputs x\_mov, y\_mov and Valid are all `0`



Here you see that he starts to receive the problematic byte again, but the WDT resets the problematic (Byte3) and we get a correct output after the clock returns to work.

## Case 11 (Reset case):



In this case we see that when we receive a reset pulse, the whole unit is Rebooted, and the mouse controller jumps to the Startup state and sends the startup data (0xF6) to confirm the connection with the output device.

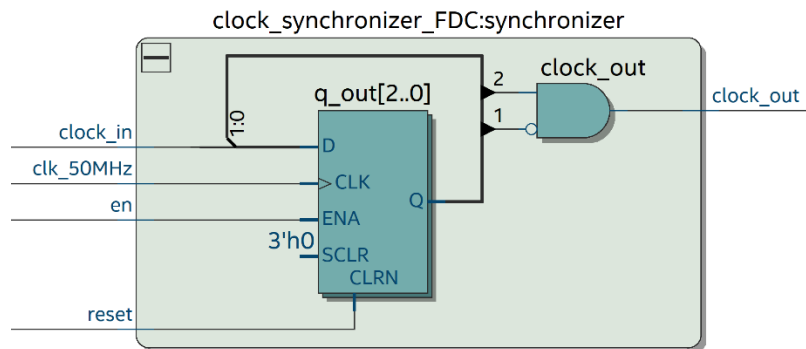
### Comment:

We Created a component named “ ParallelToSerial ” for the testbench usage only this component takes a vector of 11 and send it as serial signal on the clock rising edge when it finished its output 1



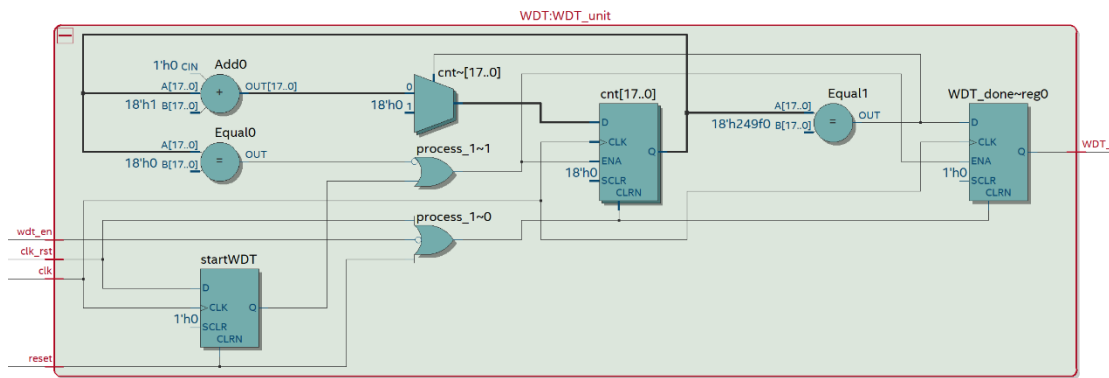
#### 4. רכיבי המערכת

##### 4.1. Clock Synchronizer



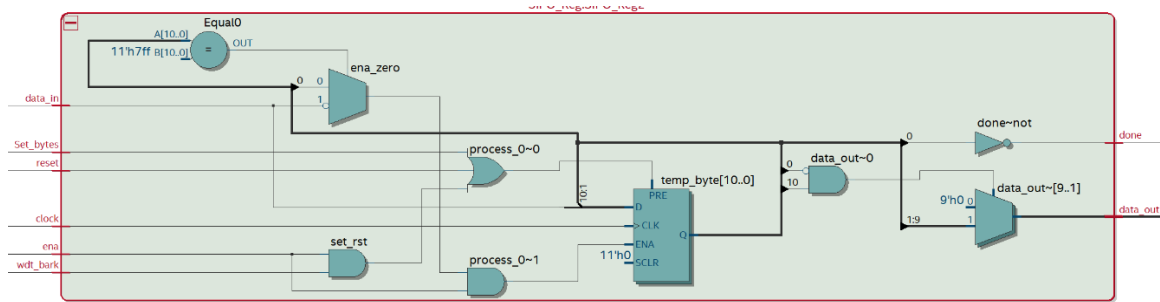
זוהי מערכת הבנויה משלושה דלגלים אשר מטרת הרכיב הינה, לסנכרון השעון שמזון (Clock\_in) יחד עם השעון של  $50[MHz]$ . תהליך הסנכרון נשלט ע"י אות ENA שמתקבל מן ה Ctrl unit, ולבסוף מוציא שעון מסונכרן אשר מזין את שאר המערכת. כמו כן קיים RESET א-סינכרוני של המערכת כולה ו מעגל לזיהויי נפילות (Fall detection circuit). היחידה מוודה שאות השעון במוצא יהיה באורך פעימת השעון של  $50[MHz]$ .

##### 4.2. Watchdog Timer (WDT)



מטרת הרכיב הינו לספור פעימות שעון במקרה שלנו עד Timeout של  $150K$  פעימות, וזאת לשם וידויי פעולה תקינה של המערכת. במילים אחרות מערכת זו הינה רשת הבטחון שלנו במקרה והרכיב היכנס למצב ERROR או במקרה ויתקיים שיבוש בפעולת המערכת. WDT הינו עושה RESET סיסטמתי לעצמו כל עוד המערכת מתפקדת כראוי, במצב שלנו כאשר הטיימר לא מתאפס בשל תקלה מסוימת הרכיב שולח אות RESET לרגיסטר אשר תקועה ומאפס אותו על מנת שיהיה אפשר לקבל את המידה החדש. פעולת ה-WDT נשלטת ע"י ENA המתקבל מן ה Ctrl unit, כמו כן קיים ליחידה RESET א-סינכרוני של המערכת כולה.

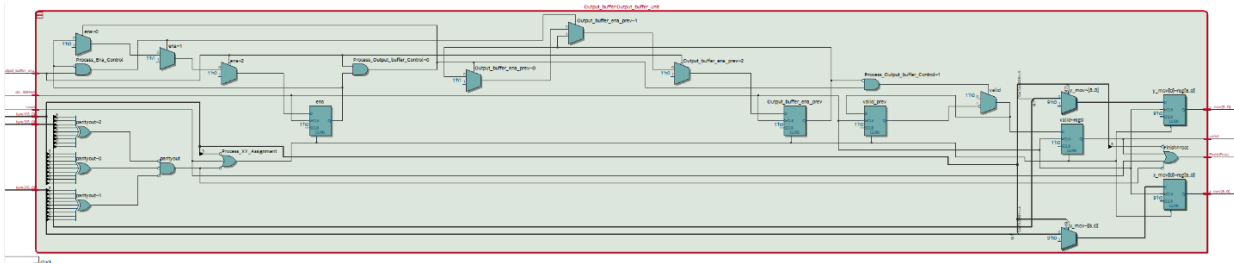
### 4.3 SIPO (Serial-In Parallel-Out) Registers



זהו רכיב של שמירת זיכרון אשר מקבל מידע רציף ושומר אותו, כאשר מוציא במוצא אותות במקביל לפי פרוטוקול מסוים שנקבע במערכת, במקרה שלנו הינו משחרר 3 אותות במקביל מכניסה רציפה אחת של מידע. רק בסיום קליטה מלאה של המידע הרציף שנכנס אנו משחררים את המידע לשלב הבא.

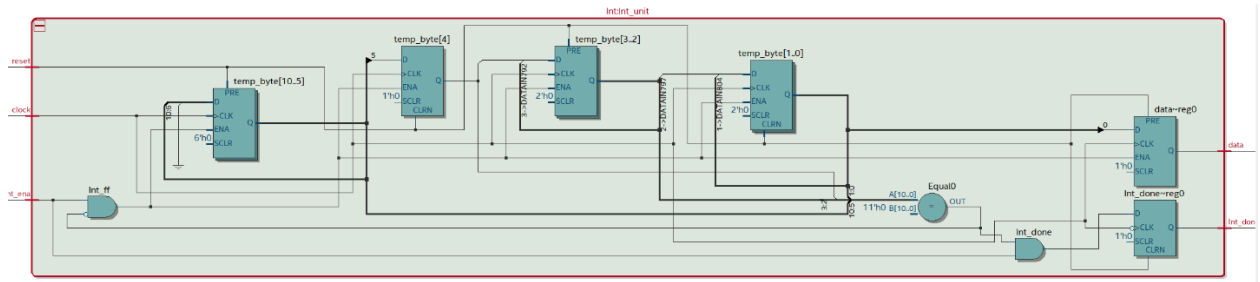
היחידה נשלטת על ידי ENA חיצוני, מערכת ביטחון של WDT, RESET א-סינכרוני כללי של המערכת ו- את של set\_byte אשר הינו אות אתחול של הרגיסטר הנשלח מן ה Ctrl unit ומטרתו לאפס את הרגיסטר כאשר אות ה- Valid חוזר חזרה ל אפס ובכך מייצג של סיום Cycle של שליחת הנתונים ביחידה.

### 4.4 Output Buffer



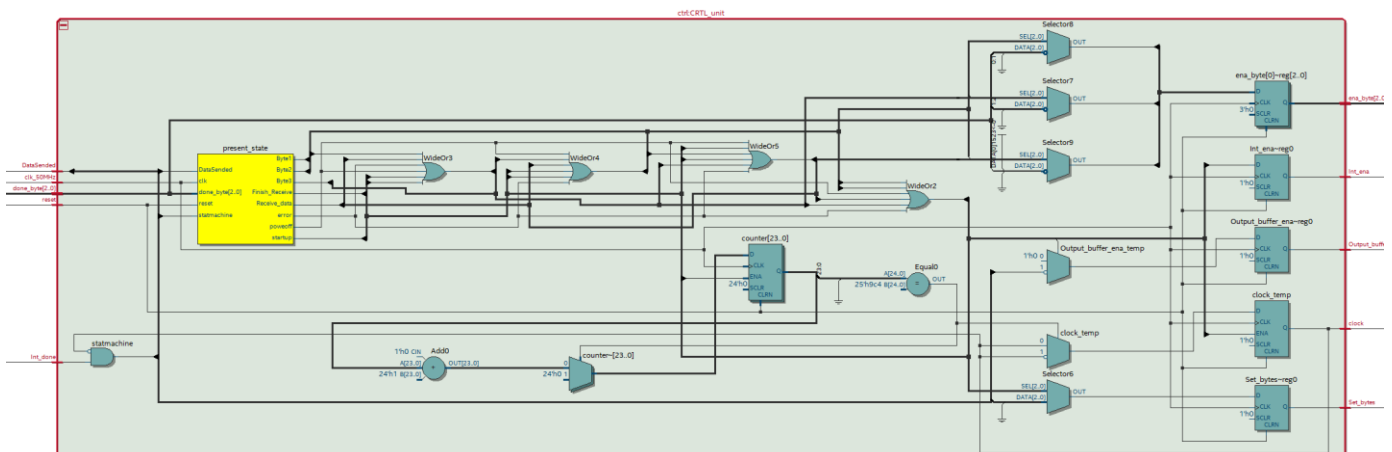
הרכיב מקבל 3 בתים כל אחד באורך 9 ביט בודקת את תקינותם, מייצרת אותות מוצא בהתאם באורך 9 ביט עם ביט הסימן במקום ביט של בדיקת תקינות הסיגנל שנכנס ו בסיום מעלה את VALID למחזור שעון אחד אשר מטרתו להודיע שהאותות הנמצאים במוצא היחידה תקינים וניתנים לקריאה. בכל מצב אחר של חוסר תקינות אות ה- VALID יישאר במצב '0', במצב של כניסה של מידע שגוי היחידה תאותחל. אלה אם היחידה קיבלה ENA חיצוני המייצג שיש צורך בקריאת התוכן המוכנס רק אז תהיה אפשרות לאות ה VALID לעבוד למחזור שעון יחיד ולא תעלה שוב עד שתהיה עלייה נוספת באות ה ENA בכניסה ליחידה, אותות ה- X\_move וה- Y\_move יעדכנו את המוצא תמיד בלי קשר לאם קיים ENA בכניסה או לא בזמן עליית שעון.

## 4.5 Init



מטרת הרכיב הינה לשלוח את אות המידע הראשוני  $0xF6$  באת הפעלת המערכת. המערכת בנויה ממספר FF אשר שומרים בתוכם ערך קבוע, אשר באת הפעלת RESET הערך הזה מיתדכן מחדש בטור ווקטור. באת קבלת ENA היחידה מייצרת מוצא של DATA עם הערך המידע  $0xF6$ . כמו כן היחידה מייצרת את של Init\_done אשר מטרתה הינה להגיד ל ctrl unit שהפעולה הסתיימה והוא יכול לעבור למצב הבא ולהוריד את אות ה- ENA חזרה ל- '0'.

## 4.6 Main Ctrl unit



מטרת הרכיב הינה לעוות כיחידת בקרה מרכזית כלומר היא בעצם מהווה את State Machine שלנו. יחידה זו היא זו ששולטת במצב המערכת מי מקבל ENA ומתי, מזהה מתי שלב אחד מסתיים ומתי צריך לעבור למצב הבא של המערכת, כמו כן ביחידה זו מיוצר השעון של  $10[KHz]$  אשר אחראי לדגימת המידע הניכנס למערכת.