

פרויקט גמר הנדסאי אלקטרוניקה

מגישות :

מרים הברי – 318344066

חני זינגר – 316286046

מנחה :

מר ברק כהן.



HOT WHEEL

תוכן

3.....	-מבוא-
4.....	-שלמי תודה-
5.....	טופס 1100 – הצעת פרויקט
6.....	-רעיון הפרויקט-
7.....	-תרשים מלבנים-
.....	-גלגל ימני- שגיאה! הסימניה אינה מוגדרת.
.....	-גלגל שמאלי- שגיאה! הסימניה אינה מוגדרת.
8.....	-הסבר תרשים המלבנים-
9.....	-תרשים חשמלי-
.....	-גלגל ימני- שגיאה! הסימניה אינה מוגדרת.
.....	-גלגל שמאלי- שגיאה! הסימניה אינה מוגדרת.
10.....	-הסבר רכיבים-
10.....	תצוגה Dot8x8
13.....	Optocoupler
15.....	Relay
16.....	חיישן טמפרטורה – LM35
16.....	ממיר מידע אנלוגי לדיגיטאלי – A/D
17.....	יחידה לייצוב מתח
19.....	-תרשים מלבנים תוכנה-
21.....	Top-Level
24.....	-Compare-
28.....	-מחלק תדר-
31.....	-בחירת מספר ביציאה-
33.....	מונה / counter
40.....	Counter_Data
41.....	-תצוגה-
57.....	-תיאור מפגשים-

-מבוא-

לכל אורגניזם בבריאה ישנה יכולת בסיסית הנקראת הוֹמֵאוֹסְטָזִיס.

כחלק מהשמירה על ההומואוסטזיס - היכולת לשמור על סביבה פנימית יציבה למרות השינויים החלים בסביבה החיצונית - מתאפיין האדם גם בתכונת ההומויאותרמיות המתבטאת ביכולתו לווסת את טמפרטורת גופו ולשמור על רמתה היציבה למרות תנאי סביבה משתנים.

טמפרטורת הגוף מושפעת מפעילות מטבולית במהלכה נפלט חום מהתאים המעלה את טמפרטורת הגוף. פעילות מטבולית מוגברת כאשר פעילות הגוף עולה. כשטמפרטורת סביבה נמוכה מטמפרטורת גוף האדם תתרחש הסעת חום מהגוף לסביבה (ול הפך). הגוף מתקן סטייה זו ע"י העלאת הפעילות המטבולית בגוף וכתוצאה יעלה חום הגוף

אולם, לאנשים בעלי מוגבלות מוטורית ישנה בדרך כלל, פעילות גופנית נמוכה. וכתוצאה מכך, כאשר טמפרטורת הסביבה נמוכה קשה יותר לגופם לעלות את הטמפרטורה בו בהתאם.

. התניידות בעזרת כיסא הגלגלים נעשית, לרוב, בעזרת הידיים שאוחזות בחישוק מתכת המחובר לגלגלי הכיסא. כתוצאה מירידת הטמפרטורה החיצונית החישוק המחובר לגלגל מאבד חום, מתקרר מאוד וכך גם הידיים האוחזות בו לזמן ממושך, דבר הגורם לירידה באיכות חיי המשתמשים. סקר קצר שערכנו בקרב בנות עם מוגבלויות העלה שזו בעיה רווחת אצל רובן.

המערכת שפיתחנו, יוצרת פתרון פרקטי שעשוי לשכלל ולשדרג את ההתניידות בכיסא הגלגלים בחודשי החורף הקרים: מערכת הכוללת גוף חימום וחיישן המווסת את טמפרטורת החישוק לחום הרצוי בתוך החישוק החלול, למרות שינויי הטמפרטורה בסביבה. וכך יקבלו המשתמשים שימוש משופר בכיסא המשמש להן כרגליים ממש.

-שלמי תודה-

"אילו פינו מלא שירה כים...אין אנחנו מספיקים להודות"
בראש ובראשונה לפני הכל תודותינו לבורא העולם, על ההזדמנות המדהימה שקבלנו ללמוד לנסות ולהחכים ולהגיע למוצר לא מבייש בכלל. תודה.

ברצוננו להודות להורים אשר תמכו בנו ובזכות עידוד רב מצדם הצלחנו להגיע להישגים גם כשחשבנו שאבד הסיכוי שנצליח לפתור בעיות שנתקלנו בהם. גם אם הינו צריכים להגיע בשעות ובימים מוזרים הם דאגו שנגיע לבית הספר כדי שנוכל להמשיך לעבוד עם הרובוט ולהגיע לתוצאות המרביות.

תודות רבות למנחה שלנו מר ברק כהן, אשר תמך בנו והיה איתנו לאורך כל הדרך הארוכה שעברנו במסגרת פרויקט זה. הוא סייע לנו בצורה יוצאת דופן בלעדיו הפרויקט הזה היה נראה אחרת לגמרי.

תודה מיוחדת לד"ר שרה גנוט שלא חסכה בכוחות ומשאבים, ע"מ שנוכל להתמודד במסלול הלימודים המשולב של האקדמיה וההנדסאים ולהצליח בשניהם.

ברצוננו להודות כמו כן לסמינר שלנו, "אופק" אשר תמך ועזר לנו להתמודד עם הקשיים של הפרויקט. ועל משאבים שלא חסך להשקיע בנו.

טופס 1100 – הצעת פרויקט

הצהרת סטודנט

שם הסטודנט: מרים הברי ת.ז. 318344066

שם הסטודנט: חני זינגר ת.ז. 318344066

שם המכללה: סמינר אופק ללימודי תואר

סמל המכללה: 440677

אני הח"מ, מצהיר בזאת כי פרויקט הגמר וספר הפרויקט המצ"ב נעשו על ידי בלבד.

פרויקט הגמר נעשה על סמך הנושאים שלמדתי במכללה ובאופן עצמאי.

פרויקט הגמר וספר הפרויקט נעשו על בסיס הנחייתו של המנחה האישי.

מקורות המידע בהם השתמשתי לביצוע פרויקט הגמר מצוינים ברשימת המקורות המצוינים בספר הפרויקט.

אני מודע לאחריות שהנני מקבל על עצמי על ידי חתימתי על הצהרה זו שכל הנאמר בה אמת ורק אמת.

חתימת הסטודנט: _____ תאריך: _____

אישור המנחה האישי

הריני מאשר שהפרויקט בוצע בהנחייתי, קראתי את ספר הפרויקט ומצאתי כי הוא מוכן לצורך הגשת הסטודנט להגנה על פרויקט גמר.

שם המנחה: _____ חתימה: _____ תאריך: _____

-רעיון הפרויקט-

בפרויקט יצרנו מערכת חימום לגלגלי כסא גלגלים.

לכל גלגל בכסא מחובר גוף חימום עם מערכת הפעלה נפרדת.

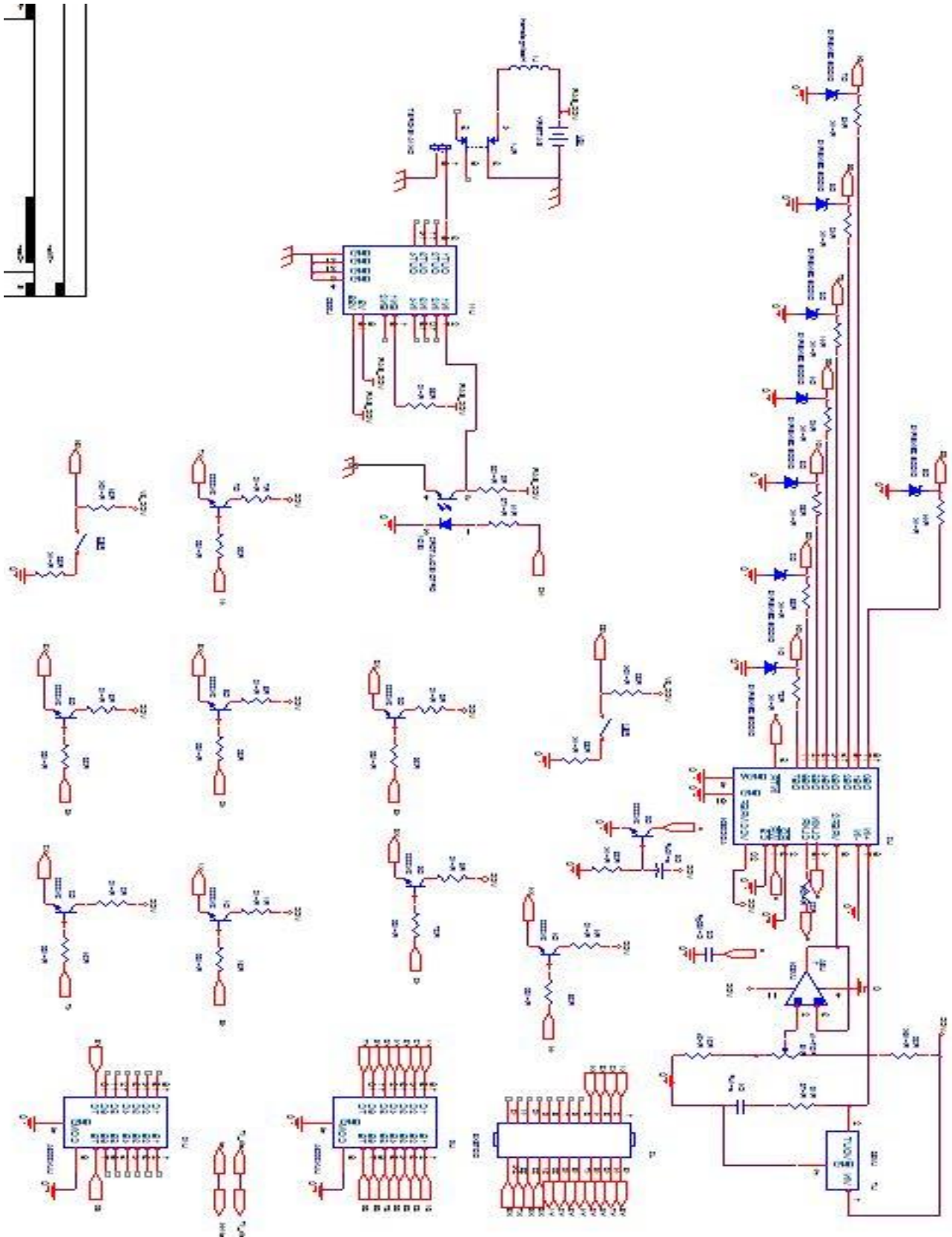
יחידת הבקרה – רכיב ה-ALTERA - שצירפנו אחראית על תפקוד המערכת. לרכיב מגיעות תוצאות חיישן הטמפרטורה המודד באופן מחזורי את הטמפרטורה על חישוקי הגלגלים המידע שמפיק החיישן עובר לרכיב A/D הממיר את המידע האנלוגי מהחיישן למידע דיגיטאלי אותו יכולה לקלוט יחידת הבקרה. לאחר שמגיע המידע ליחידת הבקרה קובע הבקר על פי התוכנה שצרבנו בו (ומפורטת להלן) בהתאם לטמפרטורה אותה בקש המשתמש בעזרת הלחצנים והטמפרטורה על חישוקי הגלגל, האם עליו להפעיל או לכבות את גוף החימום ובהתאמה איזה טמפרטורה עליו להציג על מטריצת הלדים שמציגה למשתמש את טמפרטורת החישוקים.



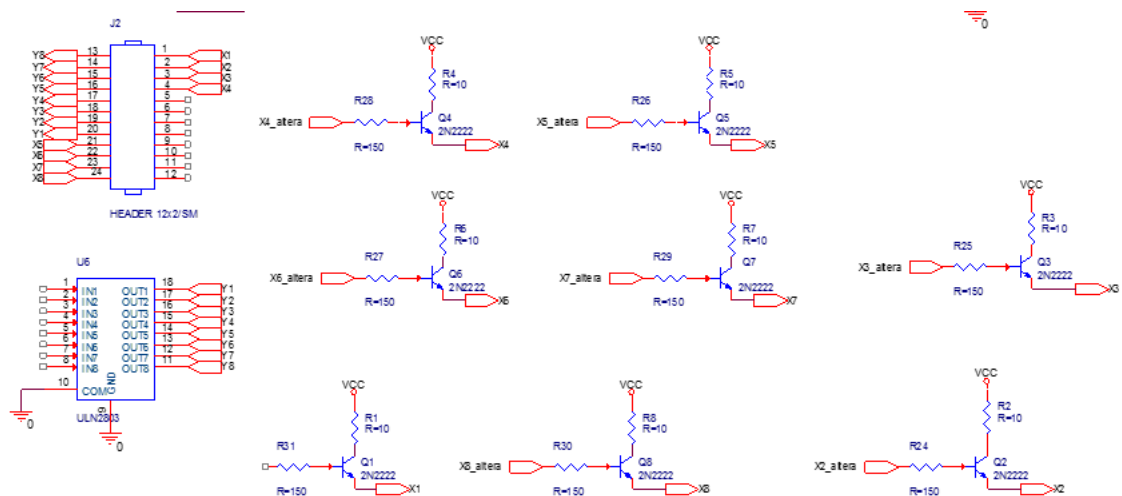
-הסבר תרשים המלבנים-

- Altera – רכיב תכן לוגי האחראי על כל פעילות המערכת. תפקידו לקבל נתונים מחיישן הטמפרטורה להפעיל ולכבות בהתאם את הגוף החימום והפעלת תצוגה.
- חיישן טמפרטורה – LM35 – חיישן זה מודד באופן מחזורי את הטמפרטורה בקשתות הגלגל, ממיר את הטמפרטורה למתח חשמלי.
- A/D – מקבל מידע בצורה אנלוגי וממיר את אותו לאות בינארי.
- מערכת הגבר מתח-זרם – מערכת טרנזיסטורים המשולבת ברכיב ULN2003 שתפקידה להגביר את הזרם הנצרך להדלקת הלדים בתצוגה.
- תצוגה – תצוגת 8X8DOT על גביה מוצגת הטמפרטורה הנמדדת בחישוקי הגלגלים.
- Opto-Coupler - רכיב אופטי המשמש לבידוד זרם בין הבקר לגוף החימום על מנת למנוע את שריפת הבקר.
- Relay – תפקידו למתג מתח וזרם גבוהים על ידי פיקוד מתח וזרם נמוכים.
- גוף החימום – דואג לחימום הגלגלים בהתאם לדרישת המשתמש.

תרשים חשמלי-



הסבר רכיבים-



תצוגה Dot8x8

ברצוננו ליצור העברת נתונים שיתקבלו מחיישן הטמפרטורה LM35 לתצוגת לדים שתיצור תשקיף לנתונים שיתקבלו. בקר הALTERA יתווך בין מוצאות החיישן לבין תצוגת הלדים והדלקת גוף החימום.

בדפי הנתונים של ALTERA נתון שזרם המוצא המקסימלי (I_{out}) שהרכיב יכול לספק הוא 25mA.

Absolute Maximum Ratings

Table 3-1 lists the absolute maximum ratings for the MAX V device family.

Table 3-1. Absolute Maximum Ratings for MAX V Devices (Note 1), (2)

Symbol	Parameter	Conditions	Minimum	Maximum	Unit
V_{CCINT}	Internal supply voltage	With respect to ground	-0.5	2.4	V
V_{CCIO}	I/O supply voltage	—	-0.5	4.6	V
V_I	DC input voltage	—	-0.5	4.6	V
I_{OUT}	DC output current, per pin	—	-25	25	mA
T_{STG}	Storage temperature	No bias	-65	150	°C
T_{AMB}	Ambient temperature	Under bias (3)	-65	135	°C
T_J	Junction temperature	TQFP and BGA packages under bias	—	135	°C

תצוגת הלדים בנויה מ-128 לדים – זוהי מטריצה בגודל 8*8 בה ממוקמים בכל מקום [i,j] שני לדים – ירוק ואדום.

הדלקת כל לד ע"פ דפי הנתונים צורכת אספקת זרם של 30mA.

על המטריצה יוצגו המעלות הנמדדות בחיישן הטמפרטורה.

A-1588BHG
Maximum Ratings

Parameter	Symbol	Value	Unit
Operating temperature	T_{OP}	-35 ~ 85	°C
Storage temperature	T_{STG}	-35 ~ 85	°C
Forward current ($T_A=25\text{ }^{\circ}\text{C}$)	I_F	30	mA per seg
Peak forward current ($T_A=25\text{ }^{\circ}\text{C}$) * ¹	I_{PF}	120	mA per seg
Reverse voltage ($T_A=25\text{ }^{\circ}\text{C}$)	V_R	5	V per seg
Power consumption ($T_A=25\text{ }^{\circ}\text{C}$)	P	80	mW per seg

*1 at 1/10 Duty Cycle

מוצאות ה-ALTERA יתחברו לשורה או עמודה בתצוגת הלדים בהתאמה. הדלקת שורה כזו תיצרוך זרם מקסימלי של 240mA - 8 לדים הצורכים כל אחד 30mA, מחוברים במקביל.

כידוע מוצא ה-ALTERA מספק זרם מקסימלי של עד 25mA בזמן שנדרש לכל שורה וטור בתצוגת הלדים 240mA כמוסבר לעיל.

הפתרון המתבקש הינו הגברת הזרם היוצא מה-ALTERA שיעשה על ידי טרנזיסטורים.

בחרנו להשתמש בטרנזיסטורים מסוג 2N2907.

זרם הבסיס בטרנזיסטור המתקבל מה-ALTERA הוא 25mA,

הגבר הטרנזיסטור לפי דפי הנתונים, $\beta = 50$.

כידוע $I_c = \beta I_b$ מה שייתן במקרה שלנו זרם $I_c = 1250\text{mA}$, כתוצאה מכך יוצרו שתי בעיות, האחת, הטרנזיסטור הנתון מוגבל בזרם I_c מקסימלי של עד 600mA, נתינת זרם גבוה מהמותר תשרוף את רכיבי הטרנזיסטור. הבעיה הנוספת היא כמוזכר לעיל - הזרם הדרוש להדלקת שורה במטריצה הוא 240mA ואילו הטרנזיסטור מספק זרם I_e גבוה בהרבה מהנדרש.

הטרנזיסטור במצב המתואר נמצא במצב רוויה.

רוויה (saturation) - טרנזיסטור נכנס למצב זה כאשר זרם הבסיס נהיה גדול מספיק והזרם בין הקולט לפולט לא יכול לגדול עוד.

לפי המאפיינים, במצב רוויה בטרנזיסטור זרם האמיטר שווה לזרם הקולקטור. כמתואר אנו רוצים לקבל זרם אמיטר של 240mA נוכל לקבל אספקת זרם זו על ידי הצבת נגד בקולקטור שתשלוט על כמות הזרם העוברת בקולקטור ומועברת לאמיטר.

חישוב גודל הנגד הדרוש:

$2.5\text{V} =$ המתח הנופל על הלדים (2.1V במקרה האידאלי) - V_{ce} (0.4V) - מתח המבוא המסופק (5V) ולכן ערך הנגד הוא: $10\Omega \sim 2.5\text{V}/240\text{mA}$.

רכיב האלטרס מספק בכניסת הטרנזיסטור מתח של 5V. המתח הנדרש בכניסת הטרנזיסטור הוא 2.1V הנדרש עבור הדלקת הלד, ו-0.7V הנופל על הדיודה בין הבסיס לאמיטר, סה"כ נדרשת הספקת מתח של 2.8V. לכן בבסיס הטרנזיסטור יש להציב נגד נוסף שעליו יפול הפרש המתחים של מתח הכניסה בטרנזיסטור והמתח המסופק (2.2V).

הזרם המסופק מהאלטרס לטרנזיסטור הוא כידוע, 25mA, בחישוב הנגד ניקח ערך נמוך יותר של

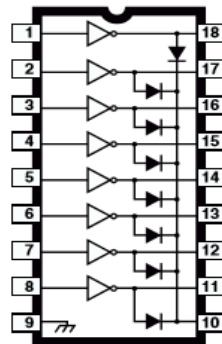
זרם (22mA) -מאמצעי בטיחות- ולכן לפי חישוב פשוט של חוק הום ערך הנגד שווה ל- 100Ω (2.2V/22mA) ולבקרת בטיחות נוספת נעלה את ערך הנגד לערך של 150Ω .

המטריצה כאמור תציג את הטמפרטורה הנמדדת בחיישן, כלומר, עליה להציג מספרים שונים על ידי הדלקת הלדים. נוכל לשלוט על הדלקת הלדים הרצויים בעזרת 16 טרנזיסטורים – 8 מהסוג המתואר לעיל מחוברים לשורות המטריצה ו-8 המשמשים כמפסקים מחוברים לטורי המטריצה בתחתית.

על מנת להדליק לד במטריצה עלינו לספק זרם לשורת הלד הרצויה על ידי הטרנזיסטור המחובר לשורה. ובנוסף לחברו בקטודה למתח של 0V כלומר למתג את הלד ב-0 או ב-1 על ידי טרנזיסטור המשמש כמפסק המחובר לטור הלד הרצוי.

לאחר מחשבה נוספת החלטנו שבכדי ליעיל ולפשט את מימוש המערכת נמיר את השימוש בטרנזיסטורים המשמשים כמפסק בשימוש ברכיב ה- 2803 thru .

למעשה במימוש הפרוייקט החלפנו את הרכיב הנ"ל ברכיב הזהה לו כמעט בכל המאפיינים מלבד חיבורים המוגדרים בצורה מעט שונה.



Dwg. No. A-10,322A

השימוש ברכיב זה פשוט למדי. על מנת להדליק לד בטור j נספק 1 לוגי ברגל ה-j. ובכך יחובר הלד הרצוי לאדמה ותתאפשר זרימת זרם בו-הדלקתו.

בקר ה-ALTEA מחובר לשלוש יחידות הפעלה שונות- האחת מטריצת הלדים המחוברת על ידי מערך טרנזיסטורים. השניה-גוף חימום המחובר על ידי שני מבודדי מתח. היחידה השלישית היא חיישן טמפרטורה וממיר אנלוגי למתח ישר-A/D.

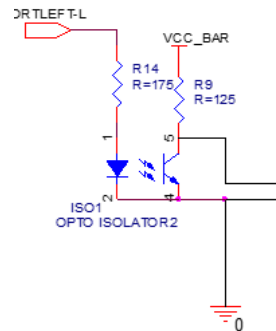
בני"ל תוארה היחידה הראשונה למרכיביה- מטריצת הלדים ומערך הטרנזיסטורים.

כעת נתאר את מרכיביה של היחידה השניה-גוף החימום.

לחיבור זה בין בקר ה-ALTEA וגוף החימום נדרשים מבודדי מתח פיזיים בדומה להרבה מקרים של חיבור רכיבים אלקטרוניים. בידוד זה נדרש משום שיתכנו פעמים בהם מסופק לרכיב יותר זרם או מתח גדולים מהנדרש לתפעולו או שנדרש מהמכשיר המספק זרם או מתח רב מיכולתו. לדוגמא גופי חימום דורשים לרוב הספק של 1000W ומתח של 220V ולפי משוואת ההפרש הזרם הנדרש הוא בערך 5A אם יחובר אותו גוף חימום לבקר ה-ALTEA שלנו שיכול לספק רק 25mA הוא ידרוש ממנו הרבה יותר זרם ממה שהוא יכול לספק ויגרום לשריפתו של רכיב ה-ALTEA. הפתרון הנדרש הוא ניתוק פיזי של בקר ה-ALTEA מגוף החימום.

את הבידוד יצרנו בעזרת רכיב ה-Optocoupler.

Optocoupler



ברכיב קיימת דיוודת לד המחוברת בקשר אופטי לטרנזיסטור, כאשר נזרים לרכיב את הזרם הדרוש להדלקתה היא תשפיע על הטרנזיסטור שיהיה בהולכה ותגרום לו לקצר זרם ומתח המוצא שווים 0. כאשר לא זרם הזרם הנדרש הטרנזיסטור בקטעון ובעקבות כך זרם המוצא שווה לא ומתח המוצא שווה לVc.

לפי דפי הנתונים:

הזרם הדרוש להדלקת הדיוידה הוא בין 10mA ל 60mA

ABSOLUTE MAXIMUM RATINGS ⁽¹⁾				
PARAMETER	TEST CONDITION	SYMBOL	VALUE	UNIT
INPUT				
Reverse voltage		V _R	5	V
Forward current		I _F	60	mA
Surge current	t ≤ 10 μs	I _{FSM}	3	A
Power dissipation		P _{diss}	100	mW
OUTPUT				
Collector emitter breakdown voltage		V _{CEO}	70	V
Emitter base breakdown voltage		V _{EB0}	7	V
Collector current		I _C	50	mA
	t ≤ 1 ms	I _C	100	mA
Power dissipation		P _{diss}	150	mW

המתח הנופל על הדיוידה הוא 1.5V.

ELECTRICAL CHARACTERISTICS ⁽¹⁾							
PARAMETER	TEST CONDITION	PART	SYMBOL	MIN.	TYP.	MAX.	UNIT
INPUT							
Forward voltage ⁽²⁾	$I_F = 50 \text{ mA}$		V_F		1.3	1.5	V
Reverse current ⁽²⁾	$V_R = 3 \text{ V}$		I_R		0.1	100	μA
Capacitance	$V_R = 0 \text{ V}$		C_O		25		pF
OUTPUT							
Collector base breakdown voltage ⁽²⁾	$I_C = 100 \mu\text{A}$		BV_{CBO}	70			V
Collector emitter breakdown voltage ⁽²⁾	$I_C = 1 \text{ mA}$		BV_{CEO}	30			V
Emitter collector breakdown voltage ⁽²⁾	$I_E = 100 \mu\text{A}$		BV_{ECO}	7			V
$I_{CEO}(\text{dark})$ ⁽²⁾	$V_{CE} = 10 \text{ V, (base open)}$	4N25			5	50	nA
		4N26			5	50	nA
		4N27			5	50	nA
		4N28			10	100	nA
$I_{CBO}(\text{dark})$ ⁽²⁾	$V_{CB} = 10 \text{ V, (emitter open)}$				2	20	nA
Collector emitter capacitance	$V_{CE} = 0$		C_{CE}		6		pF
COUPLER							
Isolation test voltage ⁽²⁾	Peak, 60 Hz		V_{IO}	5000			V
Saturation voltage, collector emitter	$I_{CE} = 2 \text{ mA, } I_F = 50 \text{ mA}$		$V_{CE(sat)}$			0.5	V
Resistance, input output ⁽²⁾	$V_{IO} = 500 \text{ V}$		R_{IO}	100			G Ω
Capacitance, input output	$f = 1 \text{ MHz}$		C_{IO}		0.6		pF

חיבור בקר ALTERA לרכיב בידוד Optocoupler:

בקר ALTERA מספק ל- Optocoupler זרם של עד 25mA ורכיב ה- Optocoupler מקבל זרם בין 10mA ל- 60mA

כדי לבצע ביטוח נוסף לבקר ALTERA שלא ישרף ניצור על ידי נגד זרם של 20mA .

לפי הנתונים הנ"ל נבחר נגד עם ערך של 175 Ω .

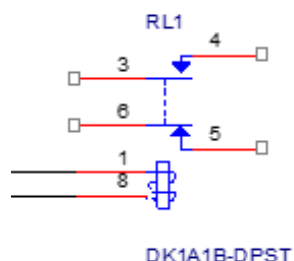
חיבור רכיב בידוד Optocoupler לרכיב הבידוד relay:

מתח זרם יעברו מרכיב בידוד Optocoupler לרכיב הבידוד relay רק כאשר זרם בדיווד 20 mA – הטרנזיסטור בקטעון.

במצב זה זרם המוצא שווה לא ומכיוון שנדרש לrelay זרם של 40mA ניצור אותו על ידי נגד

מתח V_{CC} שווה ל 5V ומכיוון שהטרנזיסטור בקטעון V_o שווה ל-0 ובעקבות כך הנגד הדרוש הוא בעל התנגדות של 125 Ω .

Relay



ברכיב זה קיים מתג המנותק פיזית מסליל וברגע שזורם בסליל זרם השראותו תגרום למיתוג המפסק ולהפעלת גוף החימום. כשלא זורם זרם המפסק פתוח.

לפי דפי הנתונים רכיב זה עובד בהספק של 0.2W ובקר ALTERA מספק לו 5V, בעקבות כך זרם הכניסה לרכיב הוא 40 mA.

Coil Parameter

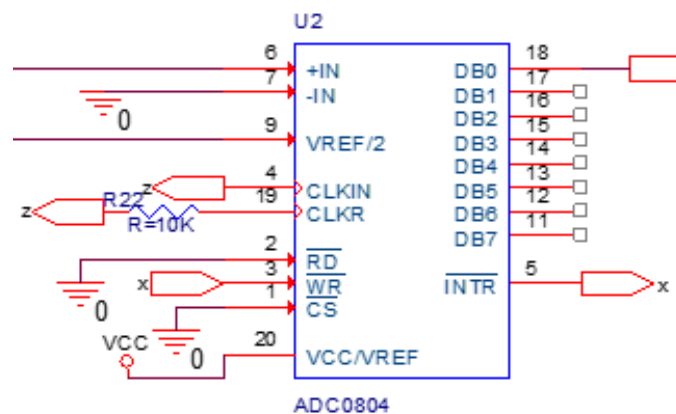
Dash numbers	Coil voltage VDC		Coil resistance $\Omega \pm 10\%$	Pickup voltage VDC(max) (75% of rated voltage)	Release voltage VDC(min) (10% of rated voltage)	Coil power consumption W	Operate Time ms	Release Time ms
	Rated	Max						
003-200	3	3.3	45	2.25	0.3	0.2	<5	<5
005-200	5	5.5	125	3.75	0.5			
006-200	6	6.6	180	4.50	0.6			
009-200	9	9.9	405	6.75	0.9			
012-200	12	13.2	720	9.00	1.2			
024-200	24	26.5	2880	18.0	2.4			
003-360	3	3.3	25	2.25	0.3	0.38	<5	<5
005-360	5	5.5	75	3.75	0.5			
006-360	6	6.6	100	4.50	0.6			
009-360	9	9.9	225	6.75	0.9			
012-360	12	13.2	400	9.00	1.2			
024-360	24	26.5	1600	18.0	2.4			
003-450	3	3.3	20	2.25	0.3	0.45	<5	<5
005-450	5	5.5	58	3.75	0.5			
006-450	6	6.6	80	4.50	0.6			
009-450	9	9.9	180	6.75	0.9			
012-450	12	13.2	320	9.00	1.2			
024-450	24	26.5	1280	18.0	2.4			

למעשה, גוף החימום יפעל אך ורק אם לא נספק לבקר הALTERA את הזרם הנדרש וברגע שנספק לו זאת הוא לא יפעל.

חיישן טמפרטורה – LM35

החיישן ממיר טמפרטורה למתח ביחס של $10\text{mV}/1^{\circ}\text{C}$. בכדי למנוע רעשי מתח במוצא החיישן נחבר אליו במקביל נגד וקבל הממתנים שינויי המתח כך שלא יקרה מצב בו יגרמו שינויים/סטיות מתח שנוצרות על ידי החיישן וישבשו את נתוני המערכת.

ממיר מידע אנלוגי לדיגיטלי – A/D



Timing Diagrams

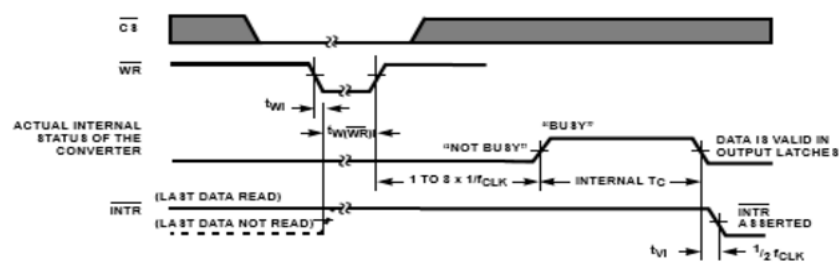


FIGURE 10A. START CONVERSION

Cs – קובע את הפעלת הרכיב.

Wr – מאשר כתיבת נתונים לרכיב.

Intr – דגל המסמל האם הרכיב קולט נתונים - הדגל ירד בקליטת נתונים ויעלה ל'1' בסיום הקליטה.

Rd – מאשר קריאת נתונים ממוצאות הרכיב.

הרגליים wr ו-intr מקוצרות זו לזו על מנת לגרום למחזוריות בקליטת הנתונים לרכיב מחיישן הטמפרטורה. שתי אלו מחוברות אף ל-reset האוטומטי לאתחול הראשוני.

Reset אוטומטי

ה-Reset האוטומטי הוא רכיב שמייצר איפוס חיצוני לרכיב או כניסה בודדת במערכת בשונה resetn רגיל המאפס את המערכת כולה.

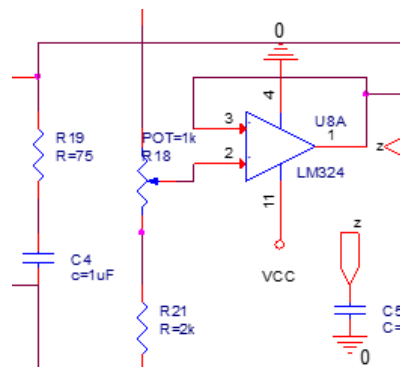
איך הוא עושה זאת?

ה-Reset האוטומטי מורכב מטרנזיסטור נגד וקבל.

ברגע הראשון המתח הנופל על הקבל הוא 0v משום שעל קבל אין קפיצות מתח. וכל המתח המסופק נופל על הנגד המקביל לכניסת הטרנזיסטור. הטרנזיסטור ברוויה ולכן המתח ב- V_A שווה ל-0v.

לאחר 5 יחידות זמן הקבל נטען ורוב המתח נופל עליו, כ-99% מהמתח נופל עליו. שארית המתח, כ-0.05% מהמתח נופל על הנגד בכניסת הטרנזיסטור. ומכיוון שמתח הבסיס קטן ממתח האמיטר הדיודה ביניהם בקטעון, וכך גם הטרנזיסטור. והמתח ב- V_A הוא '1'.

יחידה לייצוב מתח



לכניסה מספר 9 של רכיב ה-A/D נצרך מתח של 0.64V.

קיימות שתי בעיות בצריכת מתח ישירה מחיישן הטמפרטורה:

האחת, בעקבות אופי הנגדים אשר הצבנו ליצרית המתח הדרוש ההתנגדות מהם לא מדויקת ובעקבות כך ערך V_{ref} בכניסה ישתנה בהתאם.

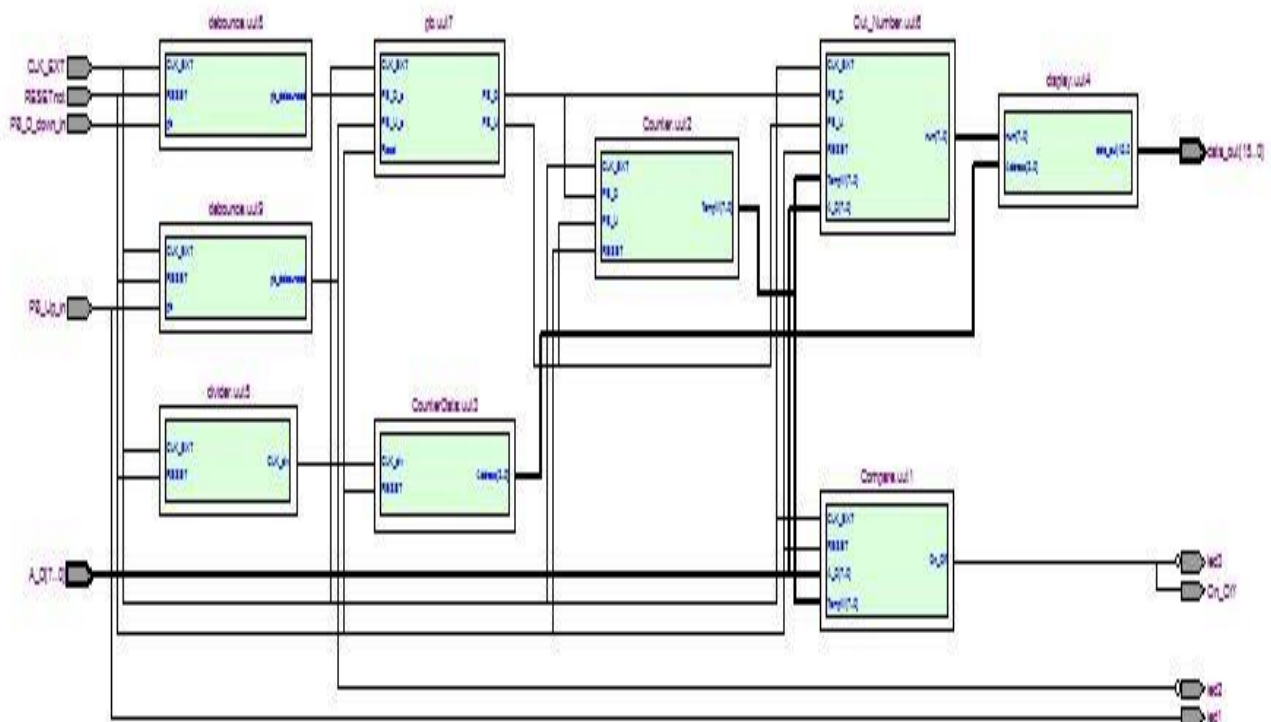
הפתרון שיצרנו לכך הוא הצבת נגד משתנה שיקבע את סכום ערכי הנגדים כך שיפיקו בדיוק את ערך המתח הרצוי.

בעיה נוספת היא שרעשים מהרכיב עלולים לשנות את V_{ref} .

הפתרון שיצרנו לבעיה זו הוא הצבת מגבר שישמור על מתח קבוע. איך? עקרון האדמה ישמור על מתח קבוע בין הכניסה ל- V_{ref} - מוצא המגבר - ומתח הכניסה למגבר.

לאחר חשיבה, ניסוי וטעייה החלפנו את יחידת הבקרה שנבחרה מלכתחילה-MAX5 ליחידה בעלת נתונים שונים המתאימים, במחשבה שניה, למערכת שלנו יותר מהקודמת. באלטרה החדשה-Cyclone2 כניסות של 3.3V ולכן מכיוון שיציאות ה-A/D מספקות ל-ALTERA 5V, לפי הנתונים, כדי לספק רק 3.3V ל-ALTERA חיברנו בין יציאות הממיר לכניסות יחידת הבקרה דיודות זנר במקביל לכניסות ה-ALTERA על מנת להכריח מתח של 3.3V בכניסה. בין דיודת הזנר ליציאת יחידת הבקרה והפרש המתחים בין המתח שמספק הממיר לבין המתח בו אנו משתמשים יפול על נגד שחיברנו בטור ביניהם.

תרשים מלבנים תוכנה-



התרשים הנ"ל מתאר את התוכנה הצרובה ביחידת הבקרה. מבואות יחידת הבקרה הם מוצאות רכיב ה-A/D ומוצאות הלחצנים. מוצאות יחידת הבקרה הם קביעת הדלקה/כיבוי של גוף החימום ותצוגת הלדים. המערכת קובעת על פי הדיאגרמה הנ"ל האם עליה להדליק את גוף החימום על מנת לקבל את הטמפרטורה הרצויה למשתמש או להפסיק את פעולתו. חיישן הטמפרטורה בודק באופן מחזורי ותדיר את הטמפרטורה הנמדדת על גבי הקשתות, את הטמפרטורה שנמדדה הוא מעביר להשוואה עם הטמפרטורה אותה ביקש המשתמש. את הטמפרטורה מעלה/מנמיך המשתמש על ידי הלחצנים החיצוניים שתוצאתם מועבר לכניסת המונה והאלגוריתם מוסיף/מחסיר על פי הפלט את ערך משתנה ה-counter בקוד, על פיו נקבעת הטמפרטורה אותה דורש המשתמש. במידה והטמפרטורות שוות – גוף החימום בטמפרטורה הרצויה – הוא מפסיק את פעולת החימום וממשיך להציג את אותה טמפרטורה על התצוגה, ולהפך, במידה והטמפרטורות שונות ממשיך גוף החימום את פעולתו ומשנה את הטמפרטורה על התצוגה בהתאם. את הטמפרטורה הרגעית מציגה המערכת על גבי התצוגה. איך זה עובד? בכדי לחסוך במשאבים קבענו שמטריצת הלדים תאיר בכל פעם רק שורה אחת ממטריצת הלדים. על התוכנית לדאוג לכך שידלקו השורות בזו אחר זו במהירות בה תדלק בכל פעם

שורה אחת ועדיין העין האנושית תקלוט את המספר הרצוי בשלמות. להדלקת השורה הרצויה דואג המונה (counter_data) שהצבנו בין התצוגה למחלק התדר. את התדר לפיו ידליק אלגוריתם התצוגה קבענו כדלהלן :

עין אנושית מסוגלת לראות כ-24 תמונות בשנייה. אנחנו רוצים להציג מספר על ידי הדלקת רצף של שורות לדים המרכיבות בסך הכללי מספר שלם. לפי הנתון אם נריץ את כל השורות בזו אחר זו תוך $1/24$ שניה נקבלת תמונה שלמה המציגה את המספר הרצוי. ולכן קבענו מחלק תדר חדש עבור התוכנית הרצויה, הוא יהיה תלוי בתדר יחידת הבקרה של המערכת ויחשב לפיו הדלקה של כל שורה למשך $1/(24*8)$ שניה.

Top-Level

```

1  library IEEE ;
2  use IEEE.std_logic_arith.all;
3  use ieee.std_logic_1164.all ;
4  use ieee.std_logic_unsigned.all ;
5
6  ENTITY Top_level IS
7  --GENERIC ( ) ;
8  PORT (
9      RESET          : IN   STD_LOGIC   ;
10     CLK_EXT         : IN   STD_LOGIC   ;
11     PB_U            : IN   STD_LOGIC   ;
12     PB_D            : IN   STD_LOGIC   ;
13     A_D             : IN   STD_LOGIC_VECTOR ( 7 downto 0 )   ;
14     data_out        : OUT  STD_LOGIC_VECTOR ( 15 downto 0 )   ;
15     On_Off          : OUT  STD_LOGIC   ;
16 ) ;
17 END entity   ;
18
19 ARCHITECTURE behave OF Top_level IS
20
21 COMPONENT Compare IS
22
23 PORT (
24     A_D          : IN   STD_LOGIC_VECTOR ( 7 downto 0 )   ;
25     Counter      : IN   STD_LOGIC_VECTOR ( 7 downto 0 )   ;
26     reset        : IN   STD_LOGIC   ;
27     CLK_EXT      : IN   STD_LOGIC   ;
28     On_Off       : out  STD_LOGIC
29 ) ;
30 END COMPONENT ;
31
32 COMPONENT Counter IS
33
34 PORT (
35     RESET          : IN   STD_LOGIC   ;
36     CLK_EXT        : IN   STD_LOGIC   ;
37     PB_U           : IN   STD_LOGIC   ;
38     PB_D           : IN   STD_LOGIC   ;
39     TempW          : out  STD_LOGIC_VECTOR ( 7 downto 0 )
40 ) ;
41 END COMPONENT ;
42
43 COMPONENT CounterData IS
44
45 PORT (
46     RESET          : IN   STD_LOGIC   ;
47     CLK_div        : IN   STD_LOGIC   ;
48     inAddress       : out  STD_LOGIC_VECTOR ( 2 downto 0 )
49 ) ;
50 END COMPONENT ;

```

```

51
52 COMPONENT display IS
53
54 PORT (
55     temp_number : in  STD_LOGIC_VECTOR ( 7 downto 0 );
56     address      : IN  STD_LOGIC_vector (2 downto 0) ;
57     data_out     : OUT STD_LOGIC_vector (15 downto 0)
58 ) ;
59 END COMPONENT ;
60
61 COMPONENT divider IS
62
63 PORT (
64     RESET      : IN  STD_LOGIC ;
65     CLK_EXT    : IN  STD_LOGIC ;
66     div        : out STD_LOGIC
67 ) ;
68 END COMPONENT ;
69
70 COMPONENT Out_Number IS
71
72 PORT (
73     RESET      : IN  STD_LOGIC ;
74     CLK_EXT    : IN  STD_LOGIC ;
75     PB_U       : IN  STD_LOGIC ;
76     PB_D       : IN  STD_LOGIC ;
77     counter_num : IN  STD_LOGIC_VECTOR ( 7 downto 0 );
78     tmp_num     : IN  STD_LOGIC_VECTOR ( 7 downto 0 );
79     num        : out STD_LOGIC_VECTOR ( 7 downto 0 )
80 ) ;
81 END COMPONENT ;
82
83 SIGNAL RESET      :STD_LOGIC ;
84 SIGNAL CLK_EXT    :STD_LOGIC ;
85 SIGNAL PB_U       :STD_LOGIC ;
86 SIGNAL PB_D       :STD_LOGIC ;
87 SIGNAL A_D        :STD_LOGIC_VECTOR ( 7 downto 0 ) ;
88 SIGNAL data_out    :STD_LOGIC_VECTOR ( 15 downto 0 ) ;
89 SIGNAL On_Off      :STD_LOGIC ;
90
91 SIGNAL Counter     :STD_LOGIC ;
92 SIGNAL CLK_div     :STD_LOGIC ;
93

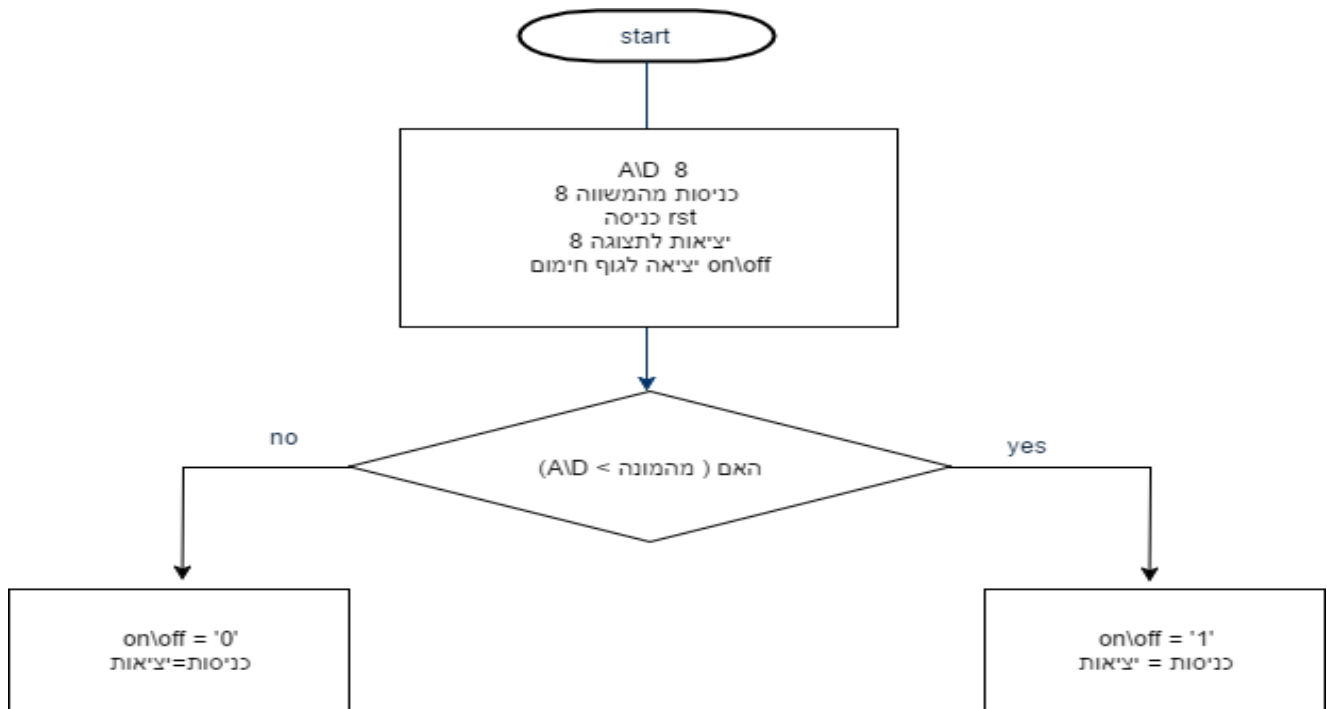
```

```

94 BEGIN
95
96 uut : Compare
97 PORT MAP (
98     A_D      => A_D ,
99     Counter  => Counter ,
100    RESET    => reset ,
101    CLK_EXT  => CLK_EXT ,
102    On_Off   => On_Off
103    ) ;
104
105 uut : Counter
106 PORT MAP (
107     RESET    => RESET ,
108     CLK_EXT  => CLK_EXT ,
109     PB_U     => PB_U ,
110     PB_D     => PB_D ,
111     TempW    => Counter
112    ) ;
113
114 uut : CounterData
115 PORT MAP (
116     RESET    => RESET ,
117     CLK_div   => CLK_div ,
118     inAddress => inAddress
119    ) ;
120
121 uut : display
122 PORT MAP (
123     temp_number => temp_number ,
124     address    => address ,
125     data_out   => data_out
126    ) ;
127
128 uut : divider
129 PORT MAP (
130     RESET    => RESET ,
131     CLK_EXT  => CLK_EXT ,
132     div      => CLK_div
133    ) ;
134
135 uut : Out_Number
136 PORT MAP (
137     RESET    => RESET ,
138     CLK_EXT  => CLK_EXT ,
139     PB_U     => PB_U ,
140     PB_D     => PB_D ,
141     Counter  => counter_num ,
142     tmp_num  => tmp_num ,
143     num      => num
144    ) ;
145
146
147 END ARCHITECTURE ;

```

-Compare-




```

1  library IEEE ;
2  use IEEE.std_logic_arith.all;
3  use ieee.std_logic_1164.all ;
4  use ieee.std_logic_unsigned.all ;
5
6  ENTITY Compare IS
7
8  --GENERIC () ;
9  PORT (
10     A_D           : IN   STD_LOGIC_VECTOR ( 7 downto 0 ) ;
11     TempW         : IN   STD_LOGIC_VECTOR ( 7 downto 0 ) ;
12     RESET         : IN   STD_LOGIC ;
13     CLK_EXT       : IN   STD_LOGIC ;
14     On_Off        : out  STD_LOGIC
15 ) ;
16 END entity ;
17
18 ARCHITECTURE behave OF Compare IS
19
20 begin
21
22 PROCESS (RESET,CLK_EXT)
23
24     begin
25
26         IF (RESET ='0')then
27
28             On_Off<='0';
29
30         elsif ( CLK_EXT'event and CLK_EXT ='1' )then
31
32             if (A_D < TempW) then
33                 On_Off <= '1';
34             else
35                 On_Off<='0';
36             end if;
37
38         end if ;
39         -- end if ;
40
41     end process ;
42
43 END behave ;
44
45
46
47

```

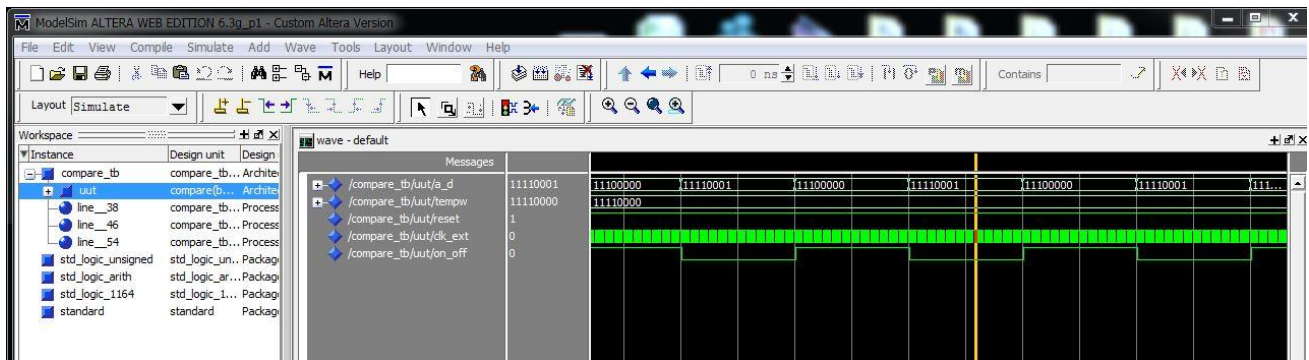
Compare Test Bench

```

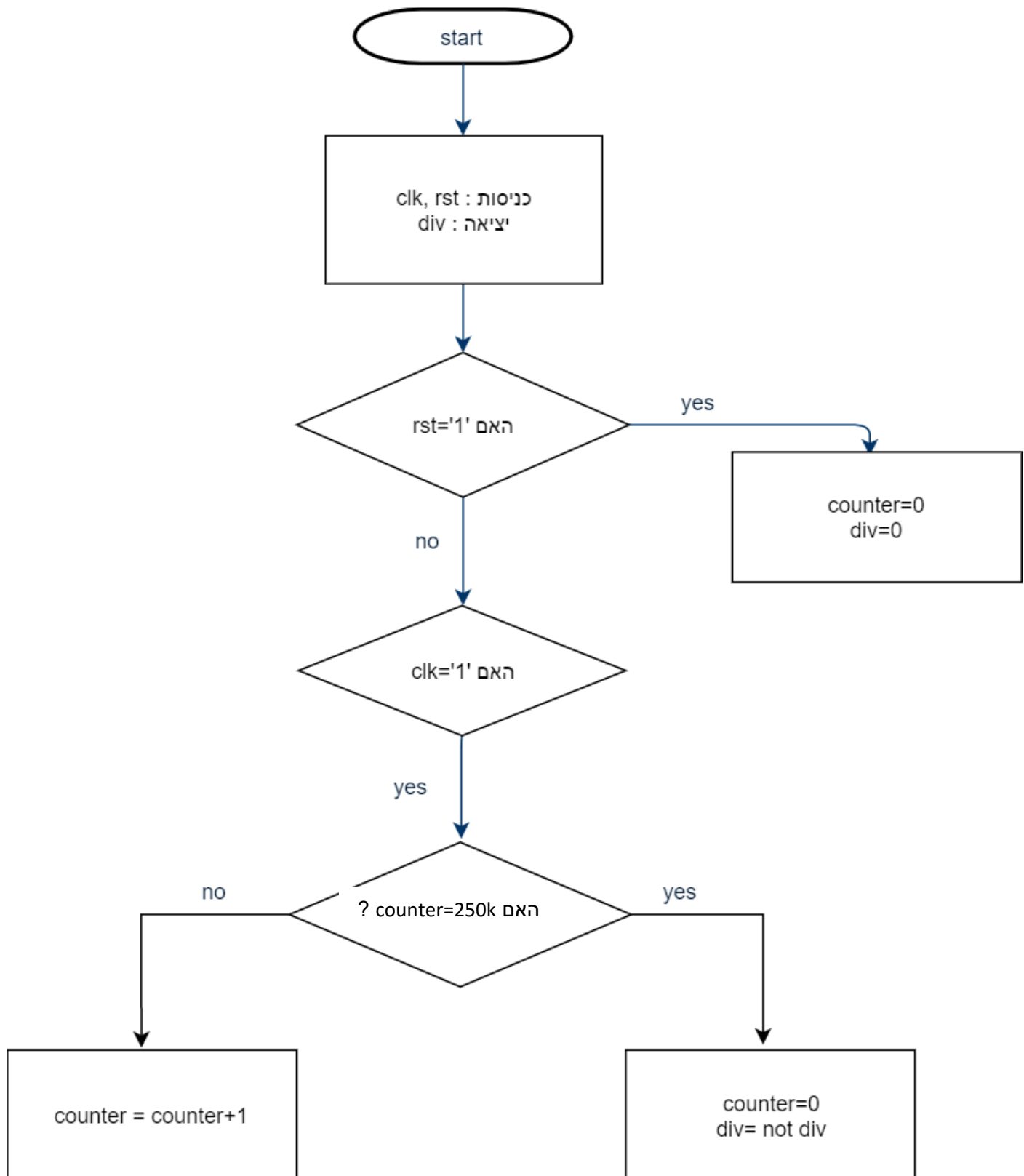
5
6 ENTITY Compare_tb IS
7   END ENTITY ;
8
9 ARCHITECTURE test_bench OF Compare_tb IS
10
11 COMPONENT Compare IS
12   PORT (
13     A_D           : IN   STD_LOGIC_VECTOR ( 7 downto 0 ) ;
14     Counter       : IN   STD_LOGIC_VECTOR ( 7 downto 0 ) ;
15     reset         : IN   STD_LOGIC ;
16     CLK_EXT       : IN   STD_LOGIC ;
17     On_Off        : out  STD_LOGIC
18   ) ;
19 END COMPONENT ;
20
21 SIGNAL A_D           :STD_LOGIC_VECTOR ( 7 downto 0 ) ;
22 SIGNAL Counter       :STD_LOGIC_VECTOR ( 7 downto 0 ) ;
23 SIGNAL reset         :STD_LOGIC ;
24 SIGNAL CLK_EXT       :STD_LOGIC ;
25 SIGNAL On_Off        :STD_LOGIC
26
27 BEGIN
28   uut : Compare
29     PORT MAP (
30       A_D      => A_D ,
31       Counter  => Counter ,
32       reset    => reset ,
33       CLK_EXT  => CLK_EXT ,
34       On_Off   => On_Off );
35
36   PROCESS
37   BEGIN
38     reset <= '1' ;
39     WAIT FOR 10 sec ;
40     reset <= '0' ;
41     WAIT FOR 10 sec ;
42   END PROCESS;
43
44   PROCESS
45   BEGIN
46     CLK_EXT <= '0' ;
47     WAIT FOR 10 ns ;
48     CLK_EXT <= '1' ;
49     WAIT FOR 10 ns ;
50   END PROCESS;
51
52   PROCESS
53   BEGIN
54     A_D <= "11110000" ;
55     Counter <= "11110000" ;
56     WAIT FOR 1 sec ;
57     A_D <= "10101010" ;
58     Counter <= "01010101" ;
59     WAIT FOR 1 sec ;
60   END PROCESS;
61 END ARCHITECTURE ;

```

בדיקת סימולציה



-מחלק תדר-



```

5      library IEEE ;
6      use IEEE.std_logic_arith.all;
7      use ieee.std_logic_1164.all ;
8      use ieee.std_logic_unsigned.all ;
9
10     ENTITY divider IS
11
12         --GENERIC () ;
13     PORT (
14         RESET           : IN    STD_LOGIC    ;
15         CLK_EXT         : IN    STD_LOGIC    ;
16         CLK_div         : OUT   STD_LOGIC
17     ) ;
18     END entity ;
19
20     ARCHITECTURE behave OF divider IS
21
22         SIGNAL Counter :STD_LOGIC_VECTOR ( 19 downto 0 ) ;
23         SIGNAL div_t   :STD_LOGIC ;
24
25     begin
26
27         --div<='0';
28
29     PROCESS (RESET,CLK_EXT)
30
31     begin
32         CLK_div <= div_t;
33
34         IF (RESET ='0')then
35
36             Counter <= x"000000";
37             div_t<='0';
38
39         elsif ( CLK_EXT'event and CLK_EXT ='1' )then
40             if(Counter= x"00220") then --250000
41                 Counter <= x"000000";
42                 div_t <= not(div_t);
43             else
44                 Counter<=Counter+1;
45             end if;
46         else
47             Counter<=Counter ;
48
49         end IF ;
50
51     END PROCESS ;
52
53
54     END behave ;

```

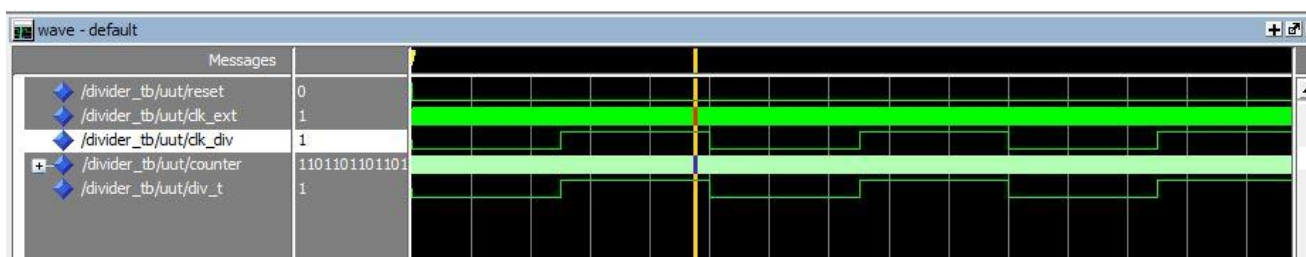
Divider Test Bench

```

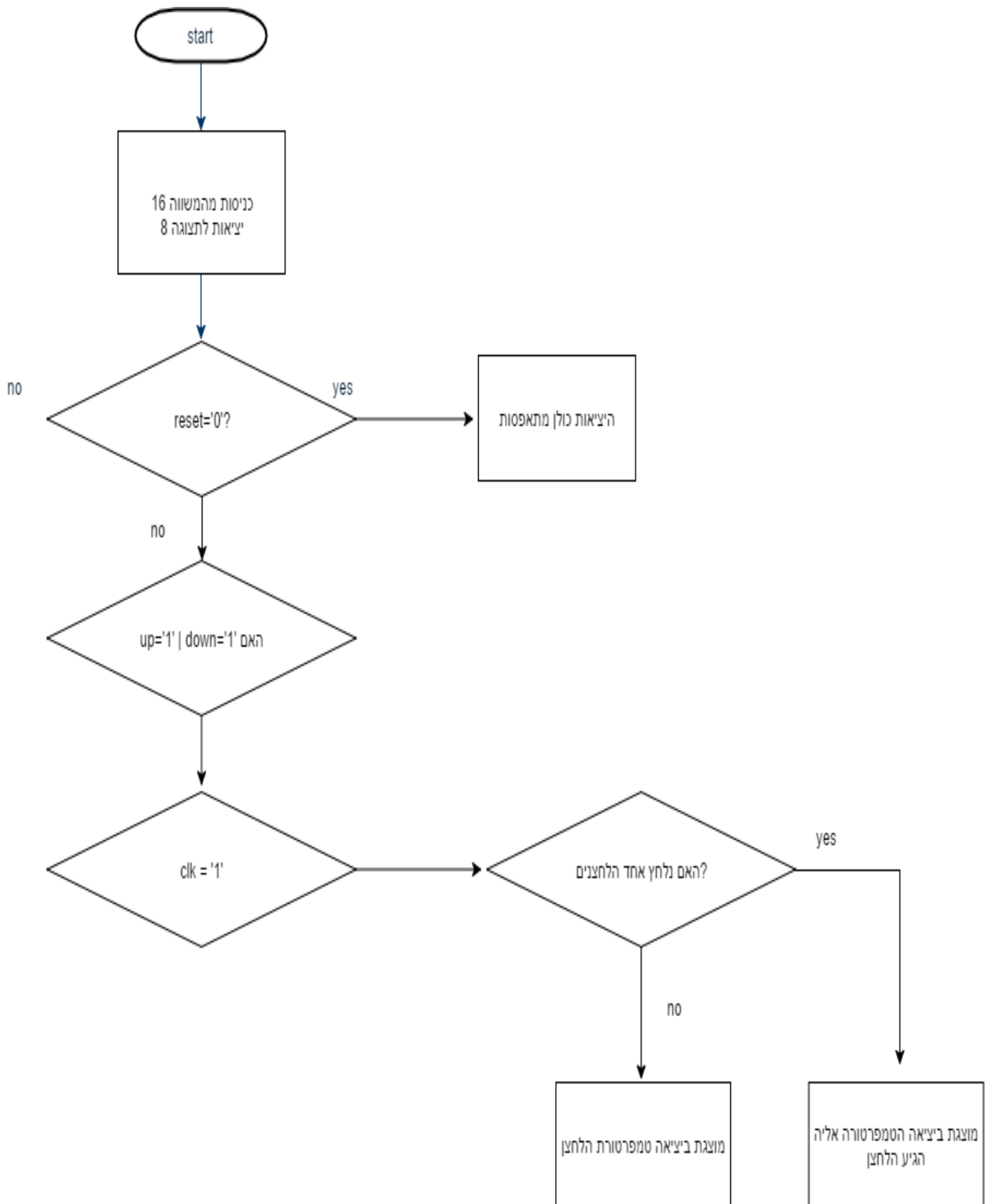
1  library IEEE ;
2  use IEEE.std_logic_arith.all;
3  use ieee.std_logic_1164.all ;
4  use ieee.std_logic_unsigned.all ;
5
6  ENTITY divider_tb IS
7  END ENTITY ;
8
9  ARCHITECTURE test_bench OF divider_tb IS
10
11  COMPONENT divider IS
12  PORT (
13      RESET      : IN  STD_LOGIC ;
14      CLK_EXT     : IN  STD_LOGIC ;
15      div        : out STD_LOGIC
16  ) ;
17  END COMPONENT ;
18
19  SIGNAL RESET      :STD_LOGIC ;
20  SIGNAL CLK_EXT     :STD_LOGIC ;
21  SIGNAL div        :STD_LOGIC ;
22
23  BEGIN
24      uut : divider
25      PORT MAP (
26          RESET      => RESET ,
27          CLK_EXT     => CLK_EXT ,
28          div        => div
29      ) ;
30
31
32      PROCESS
33      BEGIN
34          RESET <= '1' ;
35          WAIT FOR 10 sec ;
36          RESET <= '0' ;
37          WAIT FOR 10 sec ;
38      END PROCESS;
39
40      PROCESS
41      BEGIN
42          CLK_EXT <= '0' ;
43          WAIT FOR 10 ns ;
44          CLK_EXT <= '1' ;
45          WAIT FOR 10 ns ;
46      END PROCESS;
47
48  END ARCHITECTURE ;
49

```

בדיקת סימולציה



-בחירת מספר ביציאה-



```

1
2  --output number
3
4
5  library IEEE ;
6  use IEEE.std_logic_arith.all;
7  use ieee.std_logic_1164.all ;
8  use ieee.std_logic_unsigned.all ;
9
10 ENTITY Out_Number IS
11
12  --GENERIC ( ) ;
13  PORT (
14      RESET          : IN   STD_LOGIC   ;
15      CLK_EXT        : IN   STD_LOGIC   ;
16      PB_U           : IN   STD_LOGIC   ;
17      PB_D           : IN   STD_LOGIC   ;
18      TempW          : IN   STD_LOGIC_VECTOR ( 7 downto 0 );
19      A_D            : IN   STD_LOGIC_VECTOR ( 7 downto 0 );
20      num            : out  STD_LOGIC_VECTOR ( 7 downto 0 )
21  );
22  END entity ;
23
24 ARCHITECTURE behave OF Out_Number IS
25
26  SIGNAL i: STD_LOGIC_VECTOR ( 27 downto 0 );
27  SIGNAL t: STD_LOGIC;
28  SIGNAL s: STD_LOGIC;
29  SIGNAL notu: STD_LOGIC;
30  SIGNAL notd: STD_LOGIC;
31
32
33
34  begin
35
36  PROCESS (RESET,CLK_EXT)
37
38  begin
39
40
41
42      t <= (PB_U OR PB_D );
43
44      IF (RESET ='0')then
45          num <= "00000000";
46          s <='0' ;
47          i <= x"00000000" ;
48
49
50      elsif ( CLK_EXT'event and CLK_EXT ='1' )then

```



```

51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83

    if (t ='1' ) then
        s <='1' ;
        i <= x"00000000" ;
        num <= TempW;

    else

        if (s ='1' ) then

            if(i = x"5F5E100") then -- "5F5E100"
                s <='0';
                i <= x"00000000" ;
                num <= A_D;
            else
                num <= TempW;
                i <= i+1 ;

            end if ;
        else
            num <= A_D;

        end if ;

    end IF ;

end if ;

END PROCESS ;

END behave ;

```

Out_Number Test Bench

```

1  library IEEE ;
2  use IEEE.std_logic_arith.all;
3  use ieee.std_logic_1164.all ;
4  use ieee.std_logic_unsigned.all ;
5
6  ENTITY Out_Number_tb IS
7  END ENTITY ;
8
9  ARCHITECTURE test_bench OF Out_Number_tb IS
10
11  COMPONENT Out_Number IS
12  PORT (
13      RESET          : IN  STD_LOGIC ;
14      CLK_EXT        : IN  STD_LOGIC ;
15      PB_U           : IN  STD_LOGIC ;
16      PB_D           : IN  STD_LOGIC ;
17      counter_num    : IN  STD_LOGIC_VECTOR ( 7 downto 0 );
18      tmp_num        : IN  STD_LOGIC_VECTOR ( 7 downto 0 );
19      num            : out STD_LOGIC_VECTOR ( 7 downto 0 )
20  ) ;
21  END COMPONENT ;
22
23  SIGNAL RESET          :STD_LOGIC ;
24  SIGNAL CLK_EXT        :STD_LOGIC ;
25  SIGNAL PB_U           :STD_LOGIC ;
26  SIGNAL PB_D           :STD_LOGIC ;
27  SIGNAL counter_num    :STD_LOGIC_VECTOR ( 7 downto 0 );
28  SIGNAL tmp_num        :STD_LOGIC_VECTOR ( 7 downto 0 );
29  SIGNAL num            :STD_LOGIC_VECTOR ( 7 downto 0 )
30

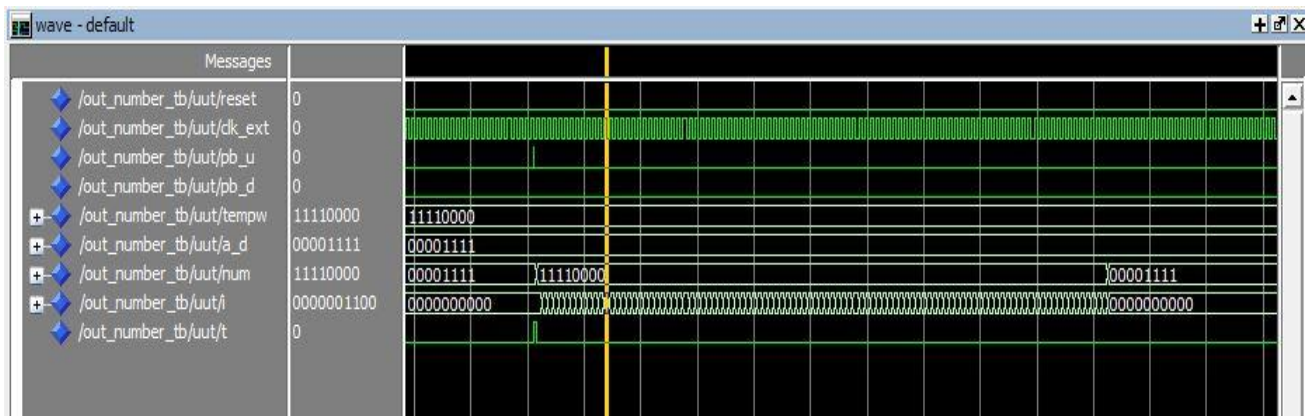
```

```

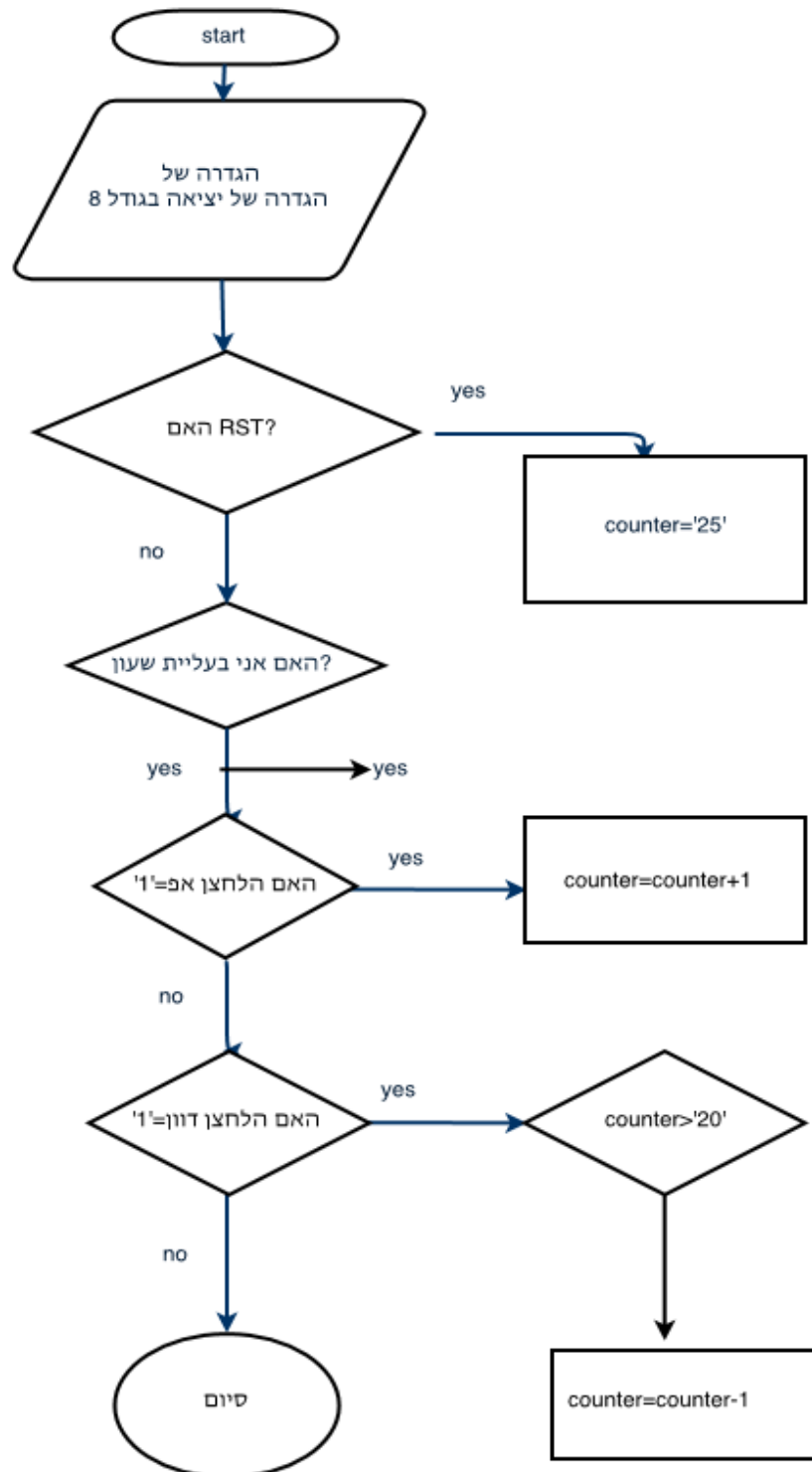
30
31 BEGIN
32 uut : Out_Number
33 PORT MAP (
34     RESET      => RESET ,
35     CLK_EXT    => CLK_EXT ,
36     PB_U       => PB_U ,
37     PB_D       => PB_D ,
38     counter_num => counter_num ,
39     tmp_num     => tmp_num ,
40     num        => num
41 );
42
43 PROCESS
44 BEGIN
45     RESET <= '1' ;
46     WAIT FOR 10 sec ;
47     RESET <= '0' ;
48     WAIT FOR 10 sec ;
49 END PROCESS;
50
51 PROCESS
52 BEGIN
53     CLK_EXT <= '0' ;
54     WAIT FOR 10 ns ;
55     CLK_EXT <= '1' ;
56     WAIT FOR 10 ns ;
57 END PROCESS;
58
59 PROCESS
60 BEGIN
61     PB_U <= '1' ;
62     WAIT FOR 3 sec ;
63     PB_D <= '1' ;
64     WAIT FOR 3 sec ;
65 END PROCESS;
66
67     counter_num <= "11110000" ;
68     tmp_num <= "00001111" ;
69
70 END ARCHITECTURE ;
71

```

בדיקת סימולציה



מונה / counter



```

1
2  library IEEE ;
3  use IEEE.std_logic_arith.all;
4  use ieee.std_logic_1164.all ;
5  use ieee.std_logic_unsigned.all ;
6
7  ENTITY Counter IS
8
9  --GENERIC ( ) ;
10 PORT (
11     RESET          : IN   STD_LOGIC   ;
12     CLK_EXT        : IN   STD_LOGIC   ;
13     PB_U           : IN   STD_LOGIC   ;
14     PB_D           : IN   STD_LOGIC   ;
15     TempW          : out  STD_LOGIC_VECTOR ( 7 downto 0 )
16 ) ;
17 END entity ;
18
19 ARCHITECTURE behave OF Counter IS
20
21     SIGNAL Counter_t : STD_LOGIC_VECTOR ( 7 downto 0 ) ;
22
23     PROCESS (RESET,PB_U,PB_D)--,CLK_EXT)
24
25     begin
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42     IF (RESET ='0')then
43         status_Up <= '0' ;
44         status_down <= '0' ;
45         Counter_d <= x"00011001" ; -- 00 ??????
46         Counter_t <= "00011001" ; -- 00 ??????
47         PB_D_d <= '0';
48         PB_U_d <= '0';
49
50
51
52
53     elsif ((CLK_EXT'event) and (CLK_EXT='1') )then
54
55         if ( (PB_U='1')) then
56             if ( Counter_t < "00101101")then --??????? 45
57                 Counter_t <= (Counter_t + '1') ;
58             end if;
59         end if;

```

```

60     if ((PB_D='1')) then
61         if ( Counter_t > "00010100")then  --??????? 20
62             Counter_t <= (Counter_t - '1') ;
63         end if;
64     end if;
65
66 else
67     Counter_t <= Counter_t ;
68 end if;
69
70
71 -- end if ;
72
73     TempW <= Counter_t;
74
75 end process ;
76
77
78 END behave ;
79
80
81

```

Counter Test Bench

```

1  library IEEE ;
2  use IEEE.std_logic_arith.all;
3  use ieee.std_logic_1164.all ;
4  use ieee.std_logic_unsigned.all ;
5
6  ENTITY Counter_tb IS
7      END ENTITY ;
8
9  ARCHITECTURE test_bench OF Counter_tb IS
10
11  COMPONENT Counter IS
12  PORT (
13      RESET           : IN   STD_LOGIC ;
14      CLK_EXT         : IN   STD_LOGIC ;
15      PB_U            : IN   STD_LOGIC ;
16      PB_D            : IN   STD_LOGIC ;
17      TempW           : out  STD_LOGIC_VECTOR ( 7 downto 0 )
18  ) ;
19  END COMPONENT ;
20
21  SIGNAL RESET        :STD_LOGIC ;
22  SIGNAL CLK_EXT      :STD_LOGIC ;
23  SIGNAL PB_U         :STD_LOGIC ;
24  SIGNAL PB_D         :STD_LOGIC ;
25  SIGNAL TempW        :STD_LOGIC_VECTOR ( 7 downto 0 )

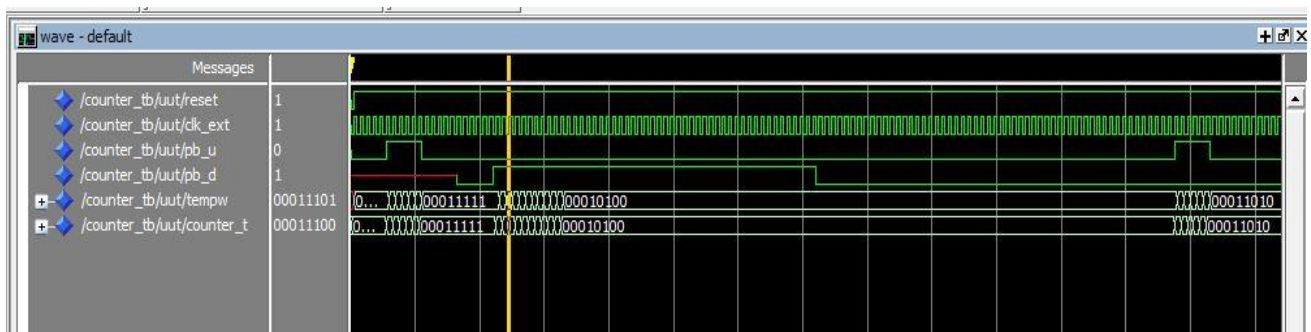
```

```

26
27 BEGIN
28 uut : Counter
29 PORT MAP (
30     RESET    => RESET ,
31     CLK_EXT  => CLK_EXT ,
32     PB_U     => PB_U ,
33     PB_D     => PB_D ,
34     TempW    => TempW ) ;
35
36 PROCESS
37 BEGIN
38     RESET <= '1' ;
39     WAIT FOR 10 sec ;
40     RESET <= '0' ;
41     WAIT FOR 10 sec ;
42 END PROCESS;
43
44 PROCESS
45 BEGIN
46     CLK_EXT <= '0' ;
47     WAIT FOR 10 ns ;
48     CLK_EXT <= '1' ;
49     WAIT FOR 10 ns ;
50 END PROCESS;
51
52 PROCESS
53 BEGIN
54     PB_U <= '1' ;
55     WAIT FOR 1 sec ;
56     PB_U <= '1' ;
57     WAIT FOR 1 sec ;
58     PB_U <= '1' ;
59     WAIT FOR 1 sec ;
60     PB_D <= '1' ;
61     WAIT FOR 1 sec ;
62     PB_D <= '1' ;
63     WAIT FOR 1 sec ;
64     PB_D <= '1' ;
65     WAIT FOR 1 sec ;
66 END PROCESS;
67
68 END ARCHITECTURE ;

```

בדיקת סימולציה



Counter_Data

```

1
2  library IEEE ;
3  use IEEE.std_logic_arith.all;
4  use ieee.std_logic_1164.all ;
5  use ieee.std_logic_unsigned.all ;
6
7  ENTITY CounterData IS
8
9      --GENERIC ( ) ;
10
11  PORT (
12      RESET          : IN    STD_LOGIC    ;
13      CLK_div         : IN    STD_LOGIC    ;
14      Address         : out   STD_LOGIC_VECTOR ( 2 doWnto 0 )
15  ) ;
16  END entity ;
17
18  ARCHITECTURE behave OF CounterData IS
19
20      SIGNAL Counter : STD_LOGIC_VECTOR ( 2 doWnto 0 ) ;
21
22  begin
23      Address<=Counter;
24
25  PROCESS (RESET,CLK_div)
26
27  begin
28
29      IF (RESET ='0')then
30
31          Counter<="000";
32
33      elsif ( CLK_div'event and CLK_div ='1' )then
34          Counter<= Counter + '1' ;
35
36      end if ;
37
38
39
40
41  end process ;
42
43      Address<=Counter;
44
45
46  END behave ;
47

```


-תצוגה-

```

1  LIBRARY ieee;
2  USE ieee.std_logic_1164.all;
3  USE ieee.std_logic_arith.all;
4  USE IEEE.std_logic_unsigned.all;
5
6
7  ENTITY display IS
8
9  PORT (
10     temp_number    : in  STD_LOGIC_VECTOR ( 7 downto 0 );
11     address         : IN  STD_LOGIC_vector (2 downto 0) ;
12     data_out        : OUT STD_LOGIC_vector (15 downto 0)
13                     )      ;
14  END display ;
15
16  ARCHITECTURE behave OF display IS
17
18
19
20     type temp00 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
21     type temp01 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
22     type temp02 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
23     type temp03 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
24     type temp04 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
25     type temp05 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
26     type temp06 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
27     type temp07 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
28     type temp08 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
29     type temp09 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
30     type temp10 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
31     type temp11 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
32     type temp12 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
33     type temp13 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
34     type temp14 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
35     type temp15 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
36     type temp16 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
37     type temp17 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
38     type temp18 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
39     type temp19 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
40     type temp20 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
41     type temp21 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
42     type temp22 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
43     type temp23 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
44     type temp24 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
45     type temp25 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
46     type temp26 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
47     type temp27 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
48     type temp28 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
49     type temp29 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
50     type temp30 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
51     type temp31 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
52     type temp32 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
53     type temp33 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);

```

```

54 type temp34 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
55 type temp35 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
56 type temp36 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
57 type temp37 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
58 type temp38 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
59 type temp39 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
60 type temp40 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
61 type temp41 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
62 type temp42 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
63 type temp43 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
64 type temp44 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
65 type temp45 is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
66 type smailly is array(0 TO 7) OF std_logic_vector (15 DOWNT0 0);
67
68 □ constant temp00_A : temp00 := (
69     "10000000    11101110"      ,
70     "01000000    10101010"      ,
71     "00100000    10101010"      ,
72     "00010000    10101010"      ,
73     "00001000    10101010"      ,
74     "00000100    10101010"      ,
75     "00000010    10101010"      ,
76     "00000001    11101110"      ,
77 );
78
79 □ constant temp01_A : temp01 := (
80     "10000000    11100100"      ,
81     "01000000    10100100"      ,
82     "00100000    10100100"      ,
83     "00010000    10100001"      ,
84     "00001000    10100001"      ,
85     "00000100    10100001"      ,
86     "00000010    10100001"      ,
87     "00000001    11100111"      ,
88 );
89
90 □ constant temp00_A : temp00 := (
91     "10000000    11101111"      ,
92     "01000000    10101001"      ,
93     "00100000    10101001"      ,
94     "00010000    10101001"      ,
95     "00001000    10101001"      ,
96     "00000100    10101001"      ,
97     "00000010    10101001"      ,
98     "00000001    11101111"      ,
99 );

```

```

100
101  constant temp00_A : temp00 := (
102      "10000000    11101111"      ,
103      "01000000    10101001"      ,
104      "00100000    10101001"      ,
105      "00010000    10101001"      ,
106      "00001000    10101001"      ,
107      "00000100    10101001"      ,
108      "00000010    10101001"      ,
109      "00000001    11101111"      ,
110  );
111
112  constant temp00_A : temp00 := (
113      "10000000    11101111"      ,
114      "01000000    10101001"      ,
115      "00100000    10101001"      ,
116      "00010000    10101001"      ,
117      "00001000    10101001"      ,
118      "00000100    10101001"      ,
119      "00000010    10101001"      ,
120      "00000001    11101111"      ,
121  );
122
123  constant temp00_A : temp00 := (
124      "10000000    11101111"      ,
125      "01000000    10101001"      ,
126      "00100000    10101001"      ,
127      "00010000    10101001"      ,
128      "00001000    10101001"      ,
129      "00000100    10101001"      ,
130      "00000010    10101001"      ,
131      "00000001    11101111"      ,
132  );
133
134  constant temp00_A : temp00 := (
135      "10000000    11101111"      ,
136      "01000000    10101001"      ,
137      "00100000    10101001"      ,
138      "00010000    10101001"      ,
139      "00001000    10101001"      ,
140      "00000100    10101001"      ,
141      "00000010    10101001"      ,
142      "00000001    11101111"      ,
143  );

```

```

144
145 = constant temp00_A : temp00 := (
146 "10000000 11101111" ,
147 "01000000 10101001" ,
148 "00100000 10101001" ,
149 "00010000 10101001" ,
150 "00001000 10101001" ,
151 "00000100 10101001" ,
152 "00000010 10101001" ,
153 "00000001 11101111"
154 );
155
156 = constant temp00_A : temp00 := (
157 "10000000 11101111" ,
158 "01000000 10101001" ,
159 "00100000 10101001" ,
160 "00010000 10101001" ,
161 "00001000 10101001" ,
162 "00000100 10101001" ,
163 "00000010 10101001" ,
164 "00000001 11101111"
165 );
166
167 = constant temp00_A : temp00 := (
168 "10000000 11101111" ,
169 "01000000 10101001" ,
170 "00100000 10101001" ,
171 "00010000 10101001" ,
172 "00001000 10101001" ,
173 "00000100 10101001" ,
174 "00000010 10101001" ,
175 "00000001 11101111"
176 );
177
178 = constant temp00_A : temp00 := (
179 "10000000 11101111" ,
180 "01000000 10101001" ,
181 "00100000 10101001" ,
182 "00010000 10101001" ,
183 "00001000 10101001" ,
184 "00000100 10101001" ,
185 "00000010 10101001" ,
186 "00000001 11101111"
187 );

```

```

188
189  constant temp00_A : temp00 := (
190      "10000000    11101111"    ,
191      "01000000    10101001"    ,
192      "00100000    10101001"    ,
193      "00010000    10101001"    ,
194      "00001000    10101001"    ,
195      "00000100    10101001"    ,
196      "00000010    10101001"    ,
197      "00000001    11101111"    ,
198  );
199
200  constant temp00_A : temp00 := (
201      "10000000    11101111"    ,
202      "01000000    10101001"    ,
203      "00100000    10101001"    ,
204      "00010000    10101001"    ,
205      "00001000    10101001"    ,
206      "00000100    10101001"    ,
207      "00000010    10101001"    ,
208      "00000001    11101111"    ,
209  );
210
211  constant temp00_A : temp00 := (
212      "10000000    11101111"    ,
213      "01000000    10101001"    ,
214      "00100000    10101001"    ,
215      "00010000    10101001"    ,
216      "00001000    10101001"    ,
217      "00000100    10101001"    ,
218      "00000010    10101001"    ,
219      "00000001    11101111"    ,
220  );
221
222  constant temp00_A : temp00 := (
223      "10000000    11101111"    ,
224      "01000000    10101001"    ,
225      "00100000    10101001"    ,
226      "00010000    10101001"    ,
227      "00001000    10101001"    ,
228      "00000100    10101001"    ,
229      "00000010    10101001"    ,
230      "00000001    11101111"    ,
231  );

```

```

233  constant temp00_A : temp00 := (
234      "10000000      11101111"      ,
235      "01000000      10101001"      ,
236      "00100000      10101001"      ,
237      "00010000      10101001"      ,
238      "00001000      10101001"      ,
239      "00000100      10101001"      ,
240      "00000010      10101001"      ,
241      "00000001      11101111"
242  );
243
244  constant temp00_A : temp00 := (
245      "10000000      11101111"      ,
246      "01000000      10101001"      ,
247      "00100000      10101001"      ,
248      "00010000      10101001"      ,
249      "00001000      10101001"      ,
250      "00000100      10101001"      ,
251      "00000010      10101001"      ,
252      "00000001      11101111"
253  );
254
255  constant temp00_A : temp00 := (
256      "10000000      11101111"      ,
257      "01000000      10101001"      ,
258      "00100000      10101001"      ,
259      "00010000      10101001"      ,
260      "00001000      10101001"      ,
261      "00000100      10101001"      ,
262      "00000010      10101001"      ,
263      "00000001      11101111"
264  );
265
266  constant temp00_A : temp00 := (
267      "10000000      11101111"      ,
268      "01000000      10101001"      ,
269      "00100000      10101001"      ,
270      "00010000      10101001"      ,
271      "00001000      10101001"      ,
272      "00000100      10101001"      ,
273      "00000010      10101001"      ,
274      "00000001      11101111"
275  );

```



```

277  constant temp00_A : temp00 := (
278      "10000000      11101111"      ,
279      "01000000      10101001"      ,
280      "00100000      10101001"      ,
281      "00010000      10101001"      ,
282      "00001000      10101001"      ,
283      "00000100      10101001"      ,
284      "00000010      10101001"      ,
285      "00000001      11101111"      ,
286  );
287
288  constant temp00_A : temp00 := (
289      "10000000      11101111"      ,
290      "01000000      10101001"      ,
291      "00100000      10101001"      ,
292      "00010000      10101001"      ,
293      "00001000      10101001"      ,
294      "00000100      10101001"      ,
295      "00000010      10101001"      ,
296      "00000001      11101111"      ,
297  );
298
299  constant temp00_A : temp00 := (
300      "10000000      11101111"      ,
301      "01000000      10101001"      ,
302      "00100000      10101001"      ,
303      "00010000      10101001"      ,
304      "00001000      10101001"      ,
305      "00000100      10101001"      ,
306      "00000010      10101001"      ,
307      "00000001      11101111"      ,
308  );
309
310  constant temp00_A : temp00 := (
311      "10000000      11101111"      ,
312      "01000000      10101001"      ,
313      "00100000      10101001"      ,
314      "00010000      10101001"      ,
315      "00001000      10101001"      ,
316      "00000100      10101001"      ,
317      "00000010      10101001"      ,
318      "00000001      11101111"      ,
319  );

```

```

320
321  constant temp00_A : temp00 := (
322      "10000000    11101111"      ,
323      "01000000    10101001"      ,
324      "00100000    10101001"      ,
325      "00010000    10101001"      ,
326      "00001000    10101001"      ,
327      "00000100    10101001"      ,
328      "00000010    10101001"      ,
329      "00000001    11101111"      ,
330  );
331
332  constant temp00_A : temp00 := (
333      "10000000    11101111"      ,
334      "01000000    10101001"      ,
335      "00100000    10101001"      ,
336      "00010000    10101001"      ,
337      "00001000    10101001"      ,
338      "00000100    10101001"      ,
339      "00000010    10101001"      ,
340      "00000001    11101111"      ,
341  );
342
343  constant temp00_A : temp00 := (
344      "10000000    11101111"      ,
345      "01000000    10101001"      ,
346      "00100000    10101001"      ,
347      "00010000    10101001"      ,
348      "00001000    10101001"      ,
349      "00000100    10101001"      ,
350      "00000010    10101001"      ,
351      "00000001    11101111"      ,
352  );
353
354  constant temp00_A : temp00 := (
355      "10000000    11101111"      ,
356      "01000000    10101001"      ,
357      "00100000    10101001"      ,
358      "00010000    10101001"      ,
359      "00001000    10101001"      ,
360      "00000100    10101001"      ,
361      "00000010    10101001"      ,
362      "00000001    11101111"      ,
363  );

```



```

364
365 constant temp00_A : temp00 := (
366 "10000000 11101111" ,
367 "01000000 10101001" ,
368 "00100000 10101001" ,
369 "00010000 10101001" ,
370 "00001000 10101001" ,
371 "00000100 10101001" ,
372 "00000010 10101001" ,
373 "00000001 11101111"
374 );
375
376 constant temp00_A : temp00 := (
377 "10000000 11101111" ,
378 "01000000 10101001" ,
379 "00100000 10101001" ,
380 "00010000 10101001" ,
381 "00001000 10101001" ,
382 "00000100 10101001" ,
383 "00000010 10101001" ,
384 "00000001 11101111"
385 );
386
387 constant temp00_A : temp00 := (
388 "10000000 11101111" ,
389 "01000000 10101001" ,
390 "00100000 10101001" ,
391 "00010000 10101001" ,
392 "00001000 10101001" ,
393 "00000100 10101001" ,
394 "00000010 10101001" ,
395 "00000001 11101111"
396 );
397
398 constant temp00_A : temp00 := (
399 "10000000 11101111" ,
400 "01000000 10101001" ,
401 "00100000 10101001" ,
402 "00010000 10101001" ,
403 "00001000 10101001" ,
404 "00000100 10101001" ,
405 "00000010 10101001" ,
406 "00000001 11101111"
407 );

```

```

408
409  = constant temp00_A : temp00 := (
410      "10000000    11101111"      ,
411      "01000000    10101001"      ,
412      "00100000    10101001"      ,
413      "00010000    10101001"      ,
414      "00001000    10101001"      ,
415      "00000100    10101001"      ,
416      "00000010    10101001"      ,
417      "00000001    11101111"      ,
418  );
419
420  = constant temp00_A : temp00 := (
421      "10000000    11101111"      ,
422      "01000000    10101001"      ,
423      "00100000    10101001"      ,
424      "00010000    10101001"      ,
425      "00001000    10101001"      ,
426      "00000100    10101001"      ,
427      "00000010    10101001"      ,
428      "00000001    11101111"      ,
429  );
430
431  = constant temp00_A : temp00 := (
432      "10000000    11101111"      ,
433      "01000000    10101001"      ,
434      "00100000    10101001"      ,
435      "00010000    10101001"      ,
436      "00001000    10101001"      ,
437      "00000100    10101001"      ,
438      "00000010    10101001"      ,
439      "00000001    11101111"      ,
440  );
441
442  = constant temp00_A : temp00 := (
443      "10000000    11101111"      ,
444      "01000000    10101001"      ,
445      "00100000    10101001"      ,
446      "00010000    10101001"      ,
447      "00001000    10101001"      ,
448      "00000100    10101001"      ,
449      "00000010    10101001"      ,
450      "00000001    11101111"      ,
451  );
452

```

```

453 constant temp00_A : temp00 := (
454     "10000000    11101111" ,
455     "01000000    10101001" ,
456     "00100000    10101001" ,
457     "00010000    10101001" ,
458     "00001000    10101001" ,
459     "00000100    10101001" ,
460     "00000010    10101001" ,
461     "00000001    11101111"
462 );
463
464 constant temp00_A : temp00 := (
465     "10000000    11101111" ,
466     "01000000    10101001" ,
467     "00100000    10101001" ,
468     "00010000    10101001" ,
469     "00001000    10101001" ,
470     "00000100    10101001" ,
471     "00000010    10101001" ,
472     "00000001    11101111"
473 );
474
475 constant temp00_A : temp00 := (
476     "10000000    11101111" ,
477     "01000000    10101001" ,
478     "00100000    10101001" ,
479     "00010000    10101001" ,
480     "00001000    10101001" ,
481     "00000100    10101001" ,
482     "00000010    10101001" ,
483     "00000001    11101111"
484 );
485
486 constant temp00_A : temp00 := (
487     "10000000    11101111" ,
488     "01000000    10101001" ,
489     "00100000    10101001" ,
490     "00010000    10101001" ,
491     "00001000    10101001" ,
492     "00000100    10101001" ,
493     "00000010    10101001" ,
494     "00000001    11101111"
495 );
496
497 constant temp00_A : temp00 := (
498     "10000000    11101111" ,
499     "01000000    10101001" ,
500     "00100000    10101001" ,
501     "00010000    10101001" ,
502     "00001000    10101001" ,
503     "00000100    10101001" ,
504     "00000010    10101001" ,
505     "00000001    11101111"
506 );
507

```

```

508 constant temp00_A : temp00 := (
509 "10000000 11101111" ,
510 "01000000 10101001" ,
511 "00100000 10101001" ,
512 "00010000 10101001" ,
513 "00001000 10101001" ,
514 "00000100 10101001" ,
515 "00000010 10101001" ,
516 "00000001 11101111"
517 );
518
519 constant temp00_A : temp00 := (
520 "10000000 11101111" ,
521 "01000000 10101001" ,
522 "00100000 10101001" ,
523 "00010000 10101001" ,
524 "00001000 10101001" ,
525 "00000100 10101001" ,
526 "00000010 10101001" ,
527 "00000001 11101111"
528 );
529
530 constant temp00_A : temp00 := (
531 "10000000 11101111" ,
532 "01000000 10101001" ,
533 "00100000 10101001" ,
534 "00010000 10101001" ,
535 "00001000 10101001" ,
536 "00000100 10101001" ,
537 "00000010 10101001" ,
538 "00000001 11101111"
539 );
540
541 constant temp00_A : temp00 := (
542 "10000000 11101111" ,
543 "01000000 10101001" ,
544 "00100000 10101001" ,
545 "00010000 10101001" ,
546 "00001000 10101001" ,
547 "00000100 10101001" ,
548 "00000010 10101001" ,
549 "00000001 11101111"
550 );
551
552 constant temp00_A : temp00 := (
553 "10000000 11101111" ,
554 "01000000 10101001" ,
555 "00100000 10101001" ,
556 "00010000 10101001" ,
557 "00001000 10101001" ,
558 "00000100 10101001" ,
559 "00000010 10101001" ,
560 "00000001 11101111"
561 );
562

```

```
563  constant temp00_A : temp00 := (  
564      "10000000      11101111"      ,  
565      "01000000      10101001"      ,  
566      "00100000      10101001"      ,  
567      "00010000      10101001"      ,  
568      "00001000      10101001"      ,  
569      "00000100      10101001"      ,  
570      "00000010      10101001"      ,  
571      "00000001      11101111"  
572  );  
573
```

```

576 BEGIN
577
578 process (address,temp_number)
579 begin
580
581 case temp_number is
582     when ("00000000") => data_out <= temp00_A(conv_integer( address(2 downto 0) ) );
583     when ("00000001") => data_out <= temp01_A(conv_integer( address(2 downto 0) ) );
584     when ("00000010") => data_out <= temp02_A(conv_integer( address(2 downto 0) ) );
585     when ("00000011") => data_out <= temp03_A(conv_integer( address(2 downto 0) ) );
586     when ("00000100") => data_out <= temp04_A(conv_integer( address(2 downto 0) ) );
587     when ("00000101") => data_out <= temp05_A(conv_integer( address(2 downto 0) ) );
588     when ("00000110") => data_out <= temp06_A(conv_integer( address(2 downto 0) ) );
589     when ("00000111") => data_out <= temp07_A(conv_integer( address(2 downto 0) ) );
590     when ("00001000") => data_out <= temp08_A(conv_integer( address(2 downto 0) ) );
591     when ("00001001") => data_out <= temp09_A(conv_integer( address(2 downto 0) ) );
592     when ("00001010") => data_out <= temp10_A(conv_integer( address(2 downto 0) ) );
593     when ("00001011") => data_out <= temp11_A(conv_integer( address(2 downto 0) ) );
594     when ("00001100") => data_out <= temp12_A(conv_integer( address(2 downto 0) ) );
595     when ("00001101") => data_out <= temp13_A(conv_integer( address(2 downto 0) ) );
596     when ("00001110") => data_out <= temp14_A(conv_integer( address(2 downto 0) ) );
597     when ("00001111") => data_out <= temp15_A(conv_integer( address(2 downto 0) ) );
598     when ("00010000") => data_out <= temp16_A(conv_integer( address(2 downto 0) ) );
599     when ("00010001") => data_out <= temp17_A(conv_integer( address(2 downto 0) ) );
600     when ("00010010") => data_out <= temp18_A(conv_integer( address(2 downto 0) ) );
601     when ("00010011") => data_out <= temp19_A(conv_integer( address(2 downto 0) ) );
602     when ("00010100") => data_out <= temp20_A(conv_integer( address(2 downto 0) ) );
603     when ("00010101") => data_out <= temp21_A(conv_integer( address(2 downto 0) ) );
604     when ("00010110") => data_out <= temp22_A(conv_integer( address(2 downto 0) ) );
605     when ("00010111") => data_out <= temp23_A(conv_integer( address(2 downto 0) ) );
606     when ("00011000") => data_out <= temp24_A(conv_integer( address(2 downto 0) ) );
607     when ("00011001") => data_out <= temp25_A(conv_integer( address(2 downto 0) ) );
608     when ("00011010") => data_out <= temp26_A(conv_integer( address(2 downto 0) ) );
609     when ("00011011") => data_out <= temp27_A(conv_integer( address(2 downto 0) ) );
610     when ("00011100") => data_out <= temp28_A(conv_integer( address(2 downto 0) ) );
611     when ("00011101") => data_out <= temp29_A(conv_integer( address(2 downto 0) ) );
612     when ("00011110") => data_out <= temp30_A(conv_integer( address(2 downto 0) ) );
613     when ("00011111") => data_out <= temp31_A(conv_integer( address(2 downto 0) ) );
614     when ("00100000") => data_out <= temp32_A(conv_integer( address(2 downto 0) ) );
615     when ("00100001") => data_out <= temp33_A(conv_integer( address(2 downto 0) ) );
616     when ("00100010") => data_out <= temp34_A(conv_integer( address(2 downto 0) ) );
617     when ("00100011") => data_out <= temp35_A(conv_integer( address(2 downto 0) ) );
618     when ("00100100") => data_out <= temp36_A(conv_integer( address(2 downto 0) ) );
619     when ("00100101") => data_out <= temp37_A(conv_integer( address(2 downto 0) ) );
620     when ("00100110") => data_out <= temp38_A(conv_integer( address(2 downto 0) ) );
621     when ("00100111") => data_out <= temp39_A(conv_integer( address(2 downto 0) ) );
622     when ("00101000") => data_out <= temp40_A(conv_integer( address(2 downto 0) ) );
623     when ("00101001") => data_out <= temp41_A(conv_integer( address(2 downto 0) ) );
624     when ("00101010") => data_out <= temp42_A(conv_integer( address(2 downto 0) ) );
625     when ("00101011") => data_out <= temp43_A(conv_integer( address(2 downto 0) ) );
626     when ("00101100") => data_out <= temp44_A(conv_integer( address(2 downto 0) ) );
627     when ("00101101") => data_out <= temp45_A(conv_integer( address(2 downto 0) ) );
628
629     when others => data_out <= smailly_mat(conv_integer( address(2 downto 0) ) );
630 end case ;
631 end process ;
632 END behave ;

```

Display Test Bench

```

1  LIBRARY ieee;
2  USE ieee.std_logic_1164.all;
3  USE ieee.std_logic_arith.all;
4  USE IEEE.std_logic_unsigned.all;
5
6  ENTITY display_tb IS
7  END ENTITY ;
8
9  ARCHITECTURE test_bench OF display_tb IS
10
11  COMPONENT display IS
12  PORT (
13    temp_number : in  STD_LOGIC_VECTOR ( 7 downto 0 );
14    address      : IN  STD_LOGIC_vector (2 downto 0) ;
15    data_out     : OUT STD_LOGIC_vector (15 downto 0)
16  ) ;
17  END COMPONENT ;
18
19  SIGNAL temp_number :STD_LOGIC_VECTOR ( 7 downto 0 );
20  SIGNAL address      :STD_LOGIC_vector (2 downto 0) ;
21  SIGNAL data_out     :STD_LOGIC_vector (15 downto 0) ;
22  SIGNAL i            :STD_LOGIC;
23
24  BEGIN
25
26    uut : display
27    PORT MAP (
28      temp_number => temp_number ,
29      address     => address ,
30      data_out    => data_out
31    ) ;
32
33    PROCESS
34    BEGIN
35      temp_number <= "00000000";
36      for i in 0 to 43 loop
37        temp_number <= temp_number+1 ;
38        WAIT FOR 1 sec ;
39      end loop;
40    END PROCESS;
41
42    PROCESS
43    BEGIN
44      address <= "00000000"
45      for i in 0 to 6 loop
46        address <= address+1 ;
47        WAIT FOR (1/24) sec ;
48      end loop;
49    END PROCESS;
50
51  END ARCHITECTURE ;
52
53

```


בדיקת סימולציה

Messages																	
/display_tb/ut/num	00000001	00000001														00000010	
/display_tb/ut/address	000	000		001	010	011	100	101	110	111	000	001					
/display_tb/ut/data_out	1000000000000000	1000000000000000...	010000001111...	001000001001...	000100001001...	000010001001...	000001001001...	000000101001...	000000011111...	100000000000...	010000...						

-תיאור מפגשים-

- מפגש 1 : תכנון הפרויקט ע"י תרשים מלבנים כללי.
- מפגש 2 : הכרת האלמנט והתצוגה.
- מפגש 3 : פתרון בעיות התצוגה ע"י הוספת רכיבים.
- מפגש 4 : הכרת חיישן הטמפרטורה.
- מפגש 5 : הכרת A/D.
- מפגש 6 : לימוד צורת חיבור חיישן הטמפרטורה לA/D.
- מפגש 7 : הכרת הרכיב האופטי.
- מפגש 8 : הכרת הרכיב הממתג.
- מפגש 9 : פתרון בעיות הרכיב האופטי והרכיב הממתג ע"י הוספת רכיבים.
- מפגש 10 : תכנון תוכנה כללית עם חלוקה ליחידות.
- מפגש 11 : יצירת תרשים זרימה ליחידת compare.
- מפגש 12 : יצירת תרשים זרימה ליחידת out_number.
- מפגש 13 : יצירת תרשים זרימה ליחידת divider.
- מפגש 14 : יצירת תרשים זרימה ליחידת counter_data.
- מפגש 15 : יצירת תרשים זרימה ליחידת display.
- מפגש 16 : יצירת תרשים זרימה ליחידת top_level.
- מפגש 17 : כתיבת התוכנה ליחידת counter.
- מפגש 18 : כתיבת התוכנה ליחידת compare.
- מפגש 19 : כתיבת התוכנה ליחידת out_number.
- מפגש 20 : כתיבת התוכנה ליחידת divider.
- מפגש 21 : כתיבת התוכנה ליחידת counter_data.
- מפגש 22 : כתיבת התוכנה ליחידת display.
- מפגש 23 : כתיבת התוכנה ליחידת top_level.
- מפגש 24 : מימוש פיזי ע"י הלחמה וחיוטים של רכיבי התצוגה.
- מפגש 25 : מימוש פיזי ע"י הלחמה וחיוטים של רכיבי חיישן הטמפרטורה ורכיב A/D.
- מפגש 26 : מימוש פיזי ע"י הלחמה וחיוטים של רכיב Relay והרכיב האופטי.
- מפגש 27 : סיכום תרשים מלבני מפורט.
- מפגש 28 : מימוש פיזי סופי כולל צריבת התוכנה.

לסיכום...

החכמנו, למדנו ונהנינו.
היה מגניב וכיף לראות את הלימודים התיאורטיים מתממשים ומתרחשים.
מחכות לפרויקט הבא...:)



7"01