חזור לתוכן עניינים

עבודת סוף התקנים לוגיים מתוכנתים

סמסטר ב' תשפ"א

שם הסטודנט, תעודת זהות, מייל: סער גוזלן , 204188403 , saar.gozlan@e.braude.ac.il
שם הסטודנט, תעודת זהות, מייל: אורן דנילוב , 203824545 , oren14dani@gmail.com

תוכן עניינים

חזור לתוכן עניינים

תיאור המערכת - TEXT CONVERT

דרישות המערכת

כללי

- המערכת קולטת מחרוזת תווים בתקשורת טורית אסינכרונית.
- המערכת משדרת מחרוזת בחזרה בתקשורת טורית אסינכרונית בסדר הפוך (מהסוף להתחלה). כלומר, המידע ישודר בסדר הפוך ביחס לסדר בו הוא נקלט.
- המערכת תופעל משעון יחיד של 27MHz.
- RESET – active low.
- המערכת תתעורר במצב idle.
- לחיצה ראשונה על הלחצן MODE תעביר את המערכת ממצב idle למצב קליטה.
- במצב קליטה, המערכת מחכה לקלוט נתונים בתקשורת טורית ממודול ה-BT. הנתונים הנקלטים ישמרו בזיכרון SRAM חיצוני ל-FPGA הקיים בערכה.
- לחיצה על הלחצן MODE תעביר את המערכת ממצב קליטה למצב שידור. במידה ואין מידע לשידור, המערכת תישאר במצב קליטה ולא תעבור למצב שידור.
- במצב שידור, המערכת תשדר את כל המידע השמור בזיכרון. בסיום שידור כל המידע, המערכת תחזור למצב קליטה ותהיה מוכנה לקלוט מידע חדש.

  לדוגמה: אם נקלטה המחרוזת "kcuL dooG", המחרוזת שתשודר, תיקלט ותוצג ע"ג הסמארטפון או הטאבלט תהיה: "Good Luck".
- מספר התווים המכסימלי שניתן לקלוט יוגדר כ- generic ובכל מקרה לא יעלה על 4095. מידע נוסף שיגיע מעבר למספר התווים המקסימלי לא יכתב לזיכרון ולא ישודר בהמשך (המידע ייזרק).
- קצב השידור והקליטה (BAUD RATE) מול המודול HC-05 יהיה 9600bps.
- על תצוגת ה-7-segment יופיע בכל רגע נתון, מספר הבתים שנקלטו ומחכים כרגע לשידור (בסיום שידור הנתונים המספר שיוצג על גבי התצוגה יהיה 0000).
- LED1 ידלוק בכל זמן שהמערכת פעילה.
- LED2 ידלוק רק כאשר המערכת במצב קליטה ויהיה כבוי בכל מצב אחר.
- LED3 ידלוק רק כאשר המערכת במצב שידור ויהיה כבוי בכל מצב אחר.
- LED4 ידלוק רק כאשר MODE לחוץ.
- רוחב המידע של הזיכרון הוא 16 ביט (2 בתים). כלומר כל כתובת בSRAM מחולקת ל2 בתים - עליון ותחתון. אנחנו ננצל רק את הבית התחתון של ערוץ המידע של הזיכרון (הבית העליון לא יהיה בשימוש).

פירוט התקשורת

- התקשורת טורית אסינכרונית כמפורט להלן:
- תקשורת טורית אסינכרונית – UART
- קצב – 9600bps
- כאשר אין תשדורת בקו הקו יהיה קבוע על 1.

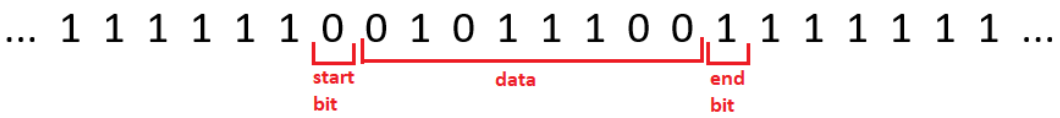
חזור לתוכן עניינים

מבנה הודעה

1 start bit of '0' , 8 data bits , 1 stop bit of '1' , no parity

למעשה כל תשדורת תפתח בstart bit ותיסגר ב end bit, וביניהם יהיו 8 ביטים של מידע.

דוגמה:



הגדרת Tbit

Tbit מגדיר את גודל כל bit מידע שמגיע אלינו או שנשלח מאיתנו בתשדורת.

$$Tbit = \frac{1}{G\_DIV} = \frac{1}{9600} = 104.166 * 10^{-6} \approx 104 \ * 10^{-6} \left[\frac{1}{Hz} = sec\right]$$

$$Tclk = \frac{1}{27M} \left[\frac{1}{Hz} = sec\right]$$

$$Tbit\ clk\ counte = \frac{Tbit}{Tclk} = \frac{\frac{1}{9600}}{\frac{1}{27M}} = \frac{27M}{9600} = 2{,}812.5 \approx \mathbf{2812}$$

כלומר, כדי לקבל אות באורך Tbit עלינו לספור 2812 עליות שעון.

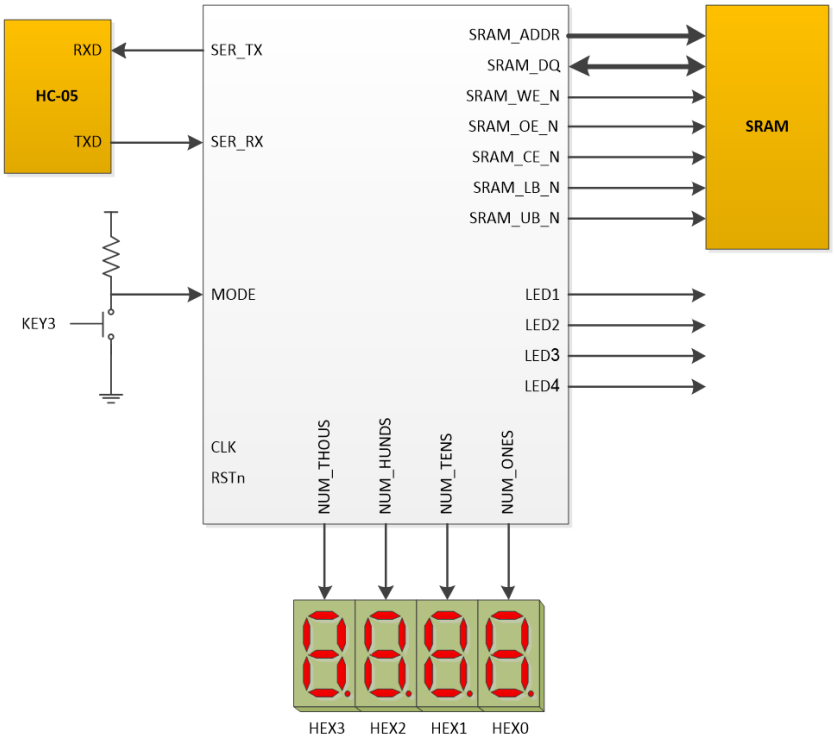
דגשים שלנו

- לכל הסיגנלים מסוג integer הוספנו range משתי סיבות:
- מטעמי אבטחת פעולת הרכיב והקלה על מציאת תקלות במערכת.
- על מנת לחסוך בחומרה (רגיסטרים) שלא בשימוש אך נובעים מהגדרת גודל הקבועים (integer למשל יכול עד 32 סיביות, אך במערכת זו יהיה שימוש ב 18 בלבד לכל היותר).
- המערכת מקבלת אות RSTn שבהגדרתו פועל כ active low וע"י שימוש בשער Not אנו הופכים אותו לאות RST שפועל כ active high.

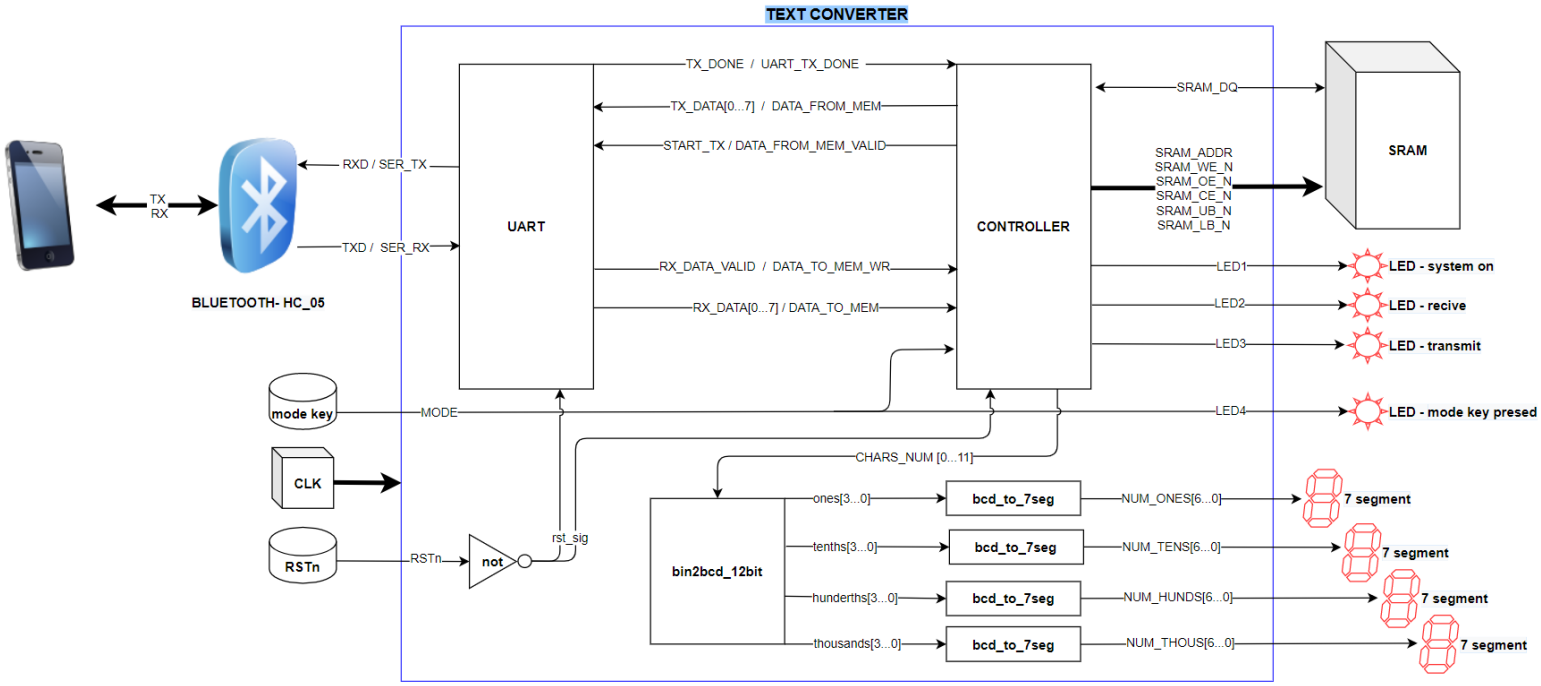
חזור לתוכן עניינים

סכמת בלוקים

כללית



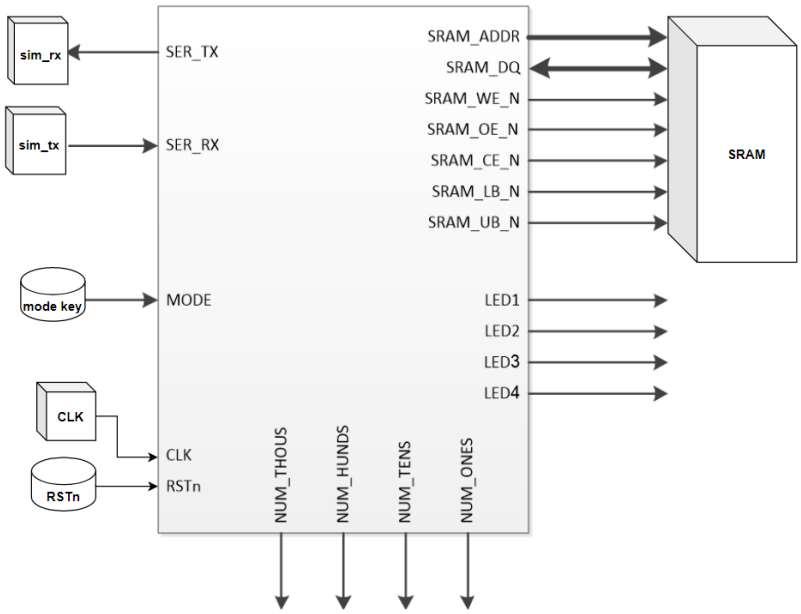
מפורטת



חזור לתוכן עניינים

סימולציה

מבנה הסימולציה :



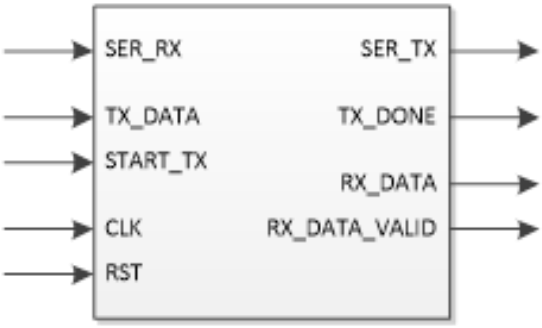
חזור לתוכן עניינים

רכיבי המערכת

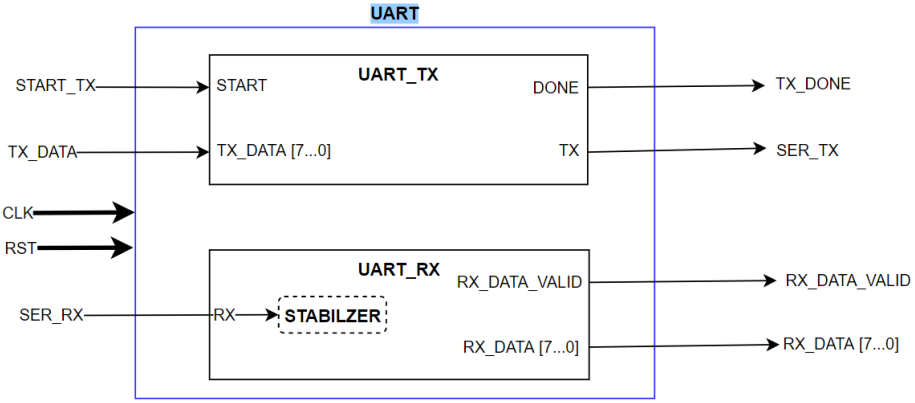
UART

סכמת בלוקים

כללית:



מפורטת:

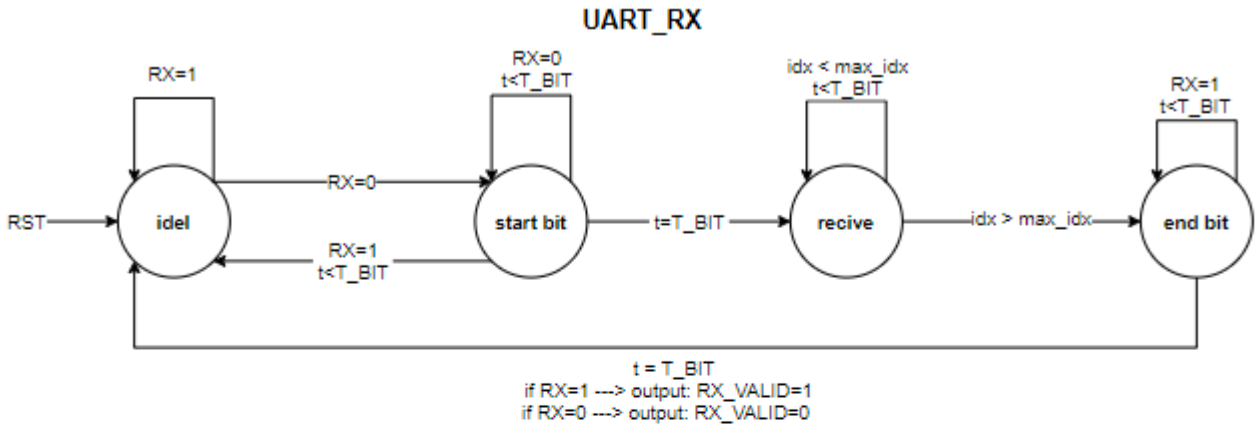


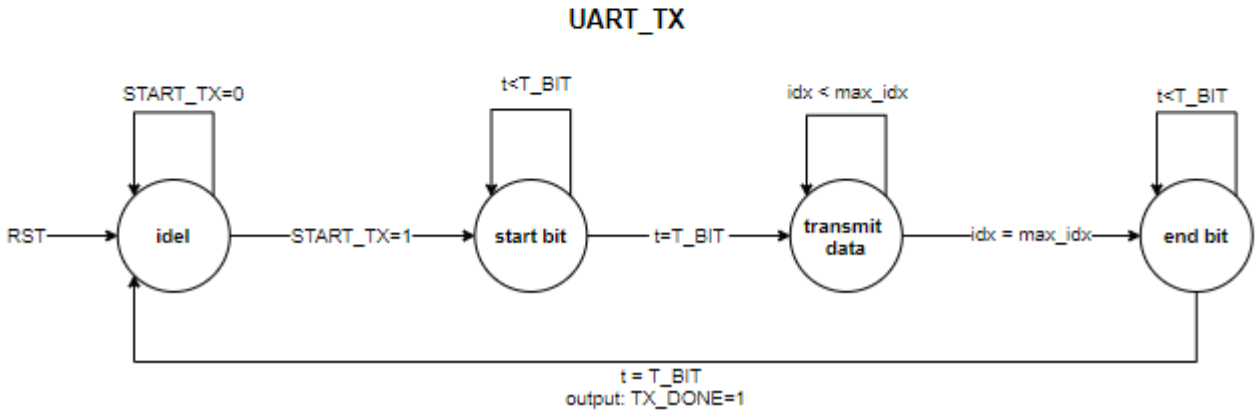
דגשים

- על מנת לעמוד בT_BIT שהוגדר השתמשנו במשתנה clk counter שלמעשה סופר עבורנו באופן מחזורי את עליות השעון ומודיע מתי לדגום הזמן הנכון לדגום.
- דגימה של אות נכנס תתבצע באמצע ה T_BIT על מנת להבטיח דגימה נכונה.

חזור לתוכן עניינים

דיאגרמת מצבים





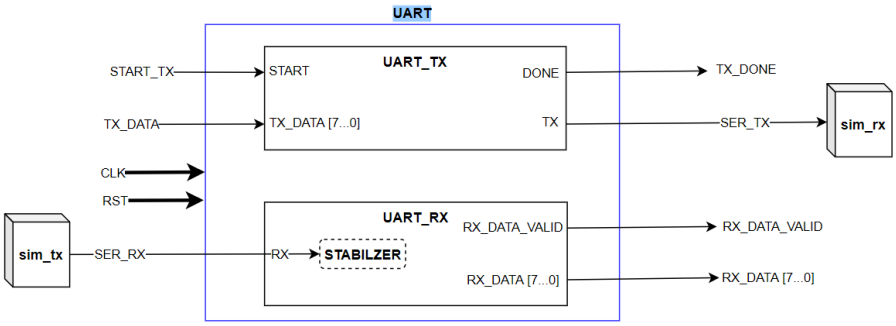
סימולציה

חיברנו את הרכיבים sim_tx , sim_rx שניתנו לנו ל URAT והרצנו test bench ב modelsim.

sim_tx מדמה לנו רכיב חיצוני ששולח אלינו תשדורת, ולכן מחובר לחוט SER_RX.

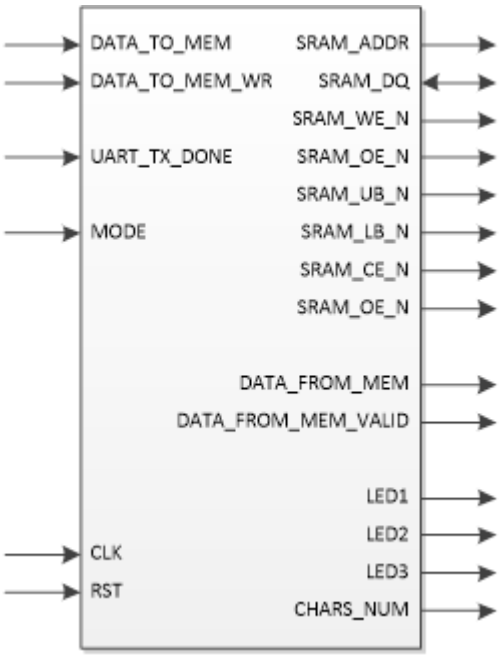
sim_rx מדמה לנו רכיב חיצוני שמקבל מאיתנו תשדורת, ולכן מחובר לחוט SER_TX.

זה נראה כך:



חזור לתוכן עניינים

CONTROLLER



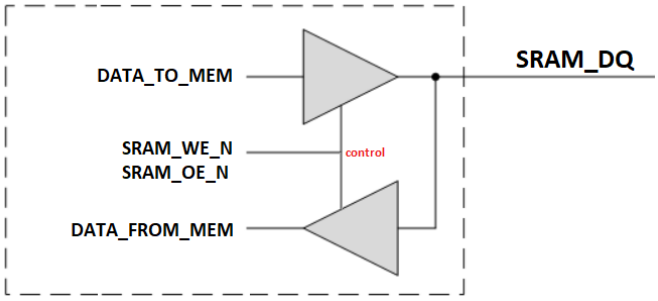
דגשים

- במצב שידור - נתעלם מלחיצה על לחצן MODE ונצא ממצב שידור רק כשסיימנו את כל מה שנמצא בזיכרון לשידור, כלומר כאשר מגיעים לכתובת 0 בזיכרון.
- בתהליך קליטה זה אפשרי בהחלט לעבור למצב שידור בין הבתים הנקלטים.

נשים לב שבמציאות כאשר ההודעה מאוד קצרה זמן התגובה מתקצר בהתאם וזה לא ריאלי שמשתמש יספיק ללחוץ על הMODE כדי לעבור לשידור. אך ככל שהקובץ הנקלט גדול יותר כך לוקח זמן רב יותר לקלוט אותו וזה מאפשר למשתמש לצאת לשידור באמצע הקליטה, כלומר יש למשתמש זמן תגובה ארוך יותר וזה ריאלי שהוא יכול ללחוץ על כפתור ה MODE (בחישוב גס להודעה הארוכה ביותר לוקח כ3.5 שניות להיקלט במערכת וזה זמן התגובה שיש למשתמש אם הוא רוצה ללחוץ פתאום על הלחצן).

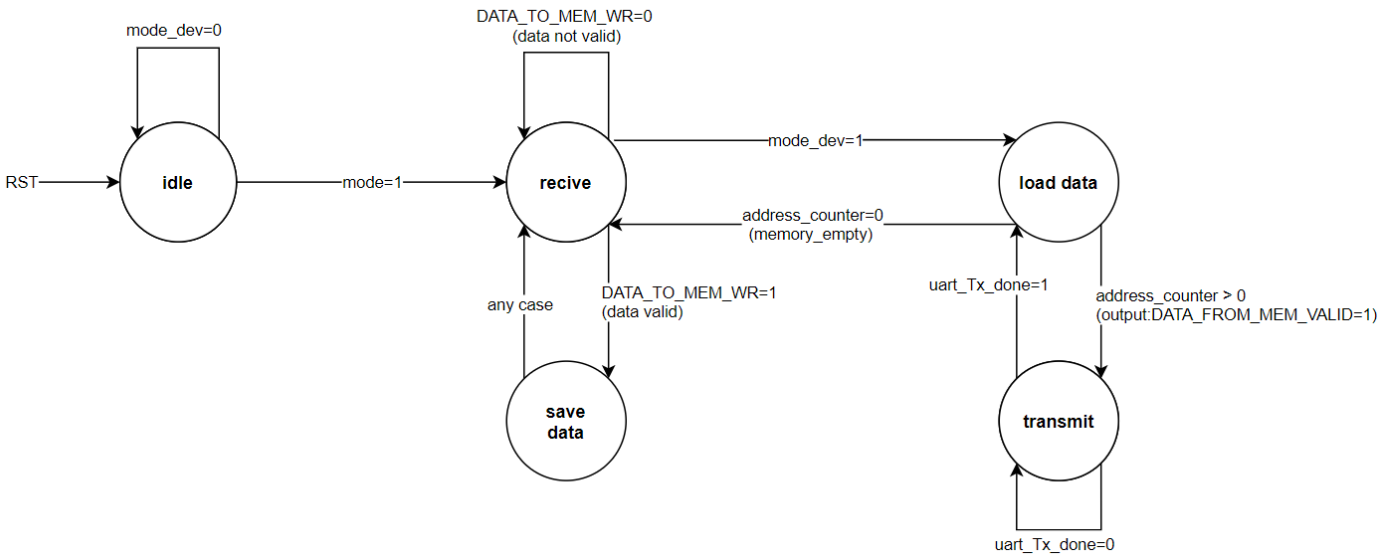
תיאורטית אפשור מצב כזה הוא בעייתי ואנחנו מודעים לו היטב, אך לצערנו דרישות המערכת אילצה אותנו לאפשר מצב זה. מצב זה יכל היה להיפתר בקלות למשל אם היה קלט נוסף מהמשתמש שבו הוא מספר כמה תווים הוא שולח לפני השליחה שלהם ואז היה ניתן להיערך בהתאם ולא לאפשר מעבר לשידור כל עוד לא נעשה אימות שנקלטו כל התווים.

- אם נגמר המקום בזיכרון – המשתמש יכול ללחוץ על ריסט על מנת לחזור לכתובת הראשונה בזיכרון ולמעשה לאפשר כתיבת לזיכרון מחדש.
- לאחר שסיימנו לשדר מידע, פשוט עוברים כתובת ולא מוחקים את המידע ששידרנו אלא משאירים את זה כ"זבל".
- קו SRAM_DQ הוא גם input וגם output. לכן עשינו שימוש three state buffer כדי למנוע מצב של קצר בין שני סיגנלים שהשתמשו בערוץ הזה, לצורך יצירת הבאפר השתמשנו בשני אותות כדי לבקר את התעבורה בערוץ.



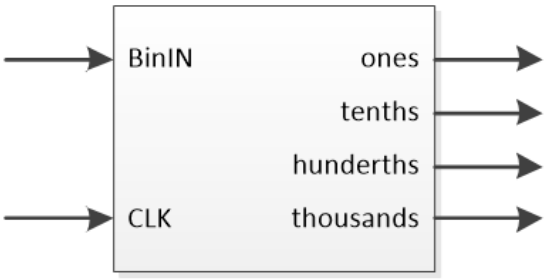
חזור לתוכן עניינים

דיאגרמת מצבים



bin2bcd_12bit

קיבלנו רכיב מוכן.



bcd_to_7seg

קיבלנו רכיב מוכן.

הרכיב מקבל אות בתצורת BCD (4 סיביות) מרכיב bin2bcd_12bit וממיר אותו לאות בתצורת 7 segment (7 סיביות) שנשלח מחוץ למערכת, ומייצגים את המספר התווים שנקלטו במערכת.

חזור לתוכן עניינים

תוצאות

משאבים

צפי שלנו לכמות הרגיסטרים הדרושים:

ניסינו לבצע הערכה לכמות הריגסטרים שאנו צופים לק:

רכיב	סוג פרמטר	פרמטר	ערך	כמות רגיסטרים ביחידה	מספר יחידות	כמות רגיסטרים
bin2bcd	רכיב			28	1	28
controller	const	max address	4095	12	1	12
controller	const	address counter	max address	12	1	12
controller	מכונת מצבים	מצבים	5	3	1	3
גוזר	רכיב			1	1	1
stabilizer	רכיב			2	2	4
text convert	const	baude rate	9600	14	1	14
uart rx	signal	rx index	7	3	1	3
uart rx	signal	clk counter	T BIT	12	1	12
uart rx	signal	max cnt	T BIT	12	1	12
uart rx	מכונת מצבים	מצבים	5	3	1	3
uart top	const	max bit	7	3	1	3
uart top	const	max bit	7	3	1	3
uart top	const	T BIT	2812	12	1	12
uart top	const	T BIT half	1406	11	1	11
uart top	רכיב			16	1	16
uart tx	signal	tx index	7	3	1	3
uart tx	signal	clk counter	T BIT	12	1	12
uart tx	מכונת מצבים	מצבים	4	2	1	2
סה"כ						**166**

משאבים בפועל

Flow Summary	
Flow Status	Successful - Mon Jul 05 21:51:16 2021
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition
Revision Name	TEXT_CONVERT
Top-level Entity Name	TEXT_CONVERT
Family	Cyclone II
Device	EP2C20F484C7
Timing Models	Final
Total logic elements	245 / 18,752 (1 %)
Total combinational functions	240 / 18,752 (1 %)
Dedicated logic registers	110 / 18,752 (< 1 %)
Total registers	110
Total pins	76 / 315 (24 %)
Total virtual pins	0
Total memory bits	0 / 239,616 (0 %)
Embedded Multiplier 9-bit elements	0 / 52 (0 %)
Total PLLs	0 / 4 (0 %)

חזור לתוכן עניינים

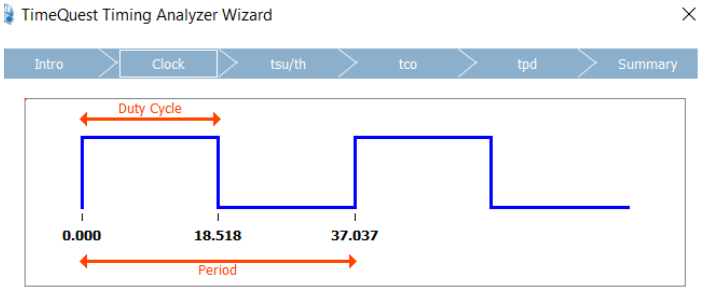
Entity	Logic Cells	Dedicated Logic Registers	I/O Registers	Memory Bits	M4Ks	DSP Elements	DSP 9x9	DSP 18x18	Pins	Virtual Pins	LUT-Only LCs	Register-Only LCs	LUT/Register LCs
Cyclone II: EP2C20F484C7													
TEXT_CONVERT	245 (0)	110 (0)	0 (0)	0	0	0	0	0	76	0	135 (0)	5 (0)	105 (0)
UART_TOP:u1_uart	99 (0)	58 (0)	0 (0)	0	0	0	0	0	0	0	41 (0)	1 (0)	57 (0)
UART_RX:u1_UART_RX	60 (58)	35 (33)	0 (0)	0	0	0	0	0	0	0	25 (25)	1 (0)	34 (34)
stabilizer:u1	2 (2)	2 (2)	0 (0)	0	0	0	0	0	0	0	0 (0)	1 (1)	1 (1)
UART_TX:u2_UART_TX	39 (39)	23 (23)	0 (0)	0	0	0	0	0	0	0	16 (16)	0 (0)	23 (23)
CONTROLLER:u2_controller	41 (38)	25 (22)	0 (0)	0	0	0	0	0	0	0	16 (16)	2 (1)	23 (22)
stabilizer:u1	2 (2)	2 (2)	0 (0)	0	0	0	0	0	0	0	0 (0)	1 (1)	1 (1)
down_der:u2	1 (1)	1 (1)	0 (0)	0	0	0	0	0	0	0	0 (0)	0 (0)	1 (1)
bin2bcd_12bit_sync:u3_bin2bcd	78 (78)	27 (27)	0 (0)	0	0	0	0	0	0	0	51 (51)	2 (2)	25 (25)
bcd_to_7seg:u4_bcd2seg_1	7 (7)	0 (0)	0 (0)	0	0	0	0	0	0	0	6 (6)	0 (0)	1 (1)
bcd_to_7seg:u5_bcd2seg_10	7 (7)	0 (0)	0 (0)	0	0	0	0	0	0	0	7 (7)	0 (0)	0 (0)
bcd_to_7seg:u6_bcd2seg_100	7 (7)	0 (0)	0 (0)	0	0	0	0	0	0	0	7 (7)	0 (0)	0 (0)
bcd_to_7seg:u7_bcd2seg_1000	7 (7)	0 (0)	0 (0)	0	0	0	0	0	0	0	7 (7)	0 (0)	0 (0)

ניתן לראות שבפועל קיבלנו פחות רגיסטרים ממה שציפינו. זה נובע ככל הנראה מכך שהקוורטוס מבצע אופטימיזציה למערכת במהלך הסינתזה.

הגדרת שעון ועמידה בתדר

הגדרת השעון בקוורטוס:

$$Tclk = \frac{1}{Fclk} = \frac{1}{27M} \approx 37.036\ [ns]$$



Specify base clock settings:

	Clock Name	Input Pin	Period	Rising	Falling
1	CLK	CLK	37.037ns	0.000	18.518

ניתוח של הקוורטוס לעמידה בתדר:

Slow Model Fmax Summary

	Fmax	Restricted Fmax	Clock Name	Note
1	93.91 MHz	93.91 MHz	CLK	

ניתן לראות שהשעון שלנו עומד בתדר.

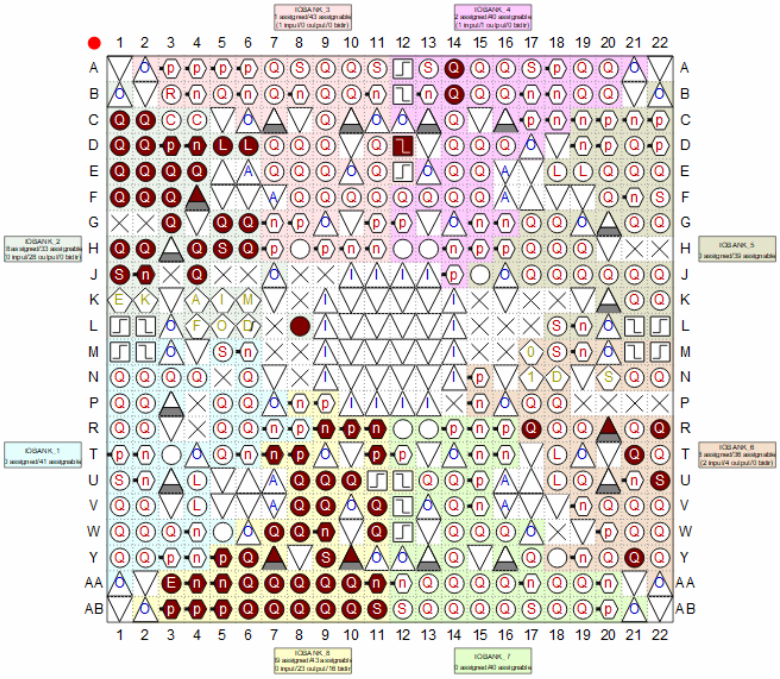
Unconstrained Paths

	Property	Setup	Hold
1	Illegal Clocks	0	0
2	Unconstrained Clocks	0	0
3	Unconstrained Input Ports	11	11
4	Unconstrained Input Port Paths	102	102
5	Unconstrained Output Ports	62	62
6	Unconstrained Output Port Paths	155	155

חזור לתוכן עניינים

הקצאת פינים

Top View - Wire Bond
Cyclone II - EP2C20F484C7



חזור לתוכן עניינים

Node Name	Direction	Location	I/O Bank
CLK	Input	PIN_D12	3
LED1	Output	PIN_R20	6
LED2	Output	PIN_U22	6
LED3	Output	PIN_R17	6
LED4	Output	PIN_Y21	6
MODE	Input	PIN_T21	6
NUM_HUNDS[6]	Output	PIN_D3	2
NUM_HUNDS[5]	Output	PIN_E4	2
NUM_HUNDS[4]	Output	PIN_E3	2
NUM_HUNDS[3]	Output	PIN_C1	2
NUM_HUNDS[2]	Output	PIN_C2	2
NUM_HUNDS[1]	Output	PIN_G6	2
NUM_HUNDS[0]	Output	PIN_G5	2
NUM_ONES[6]	Output	PIN_E2	2
NUM_ONES[5]	Output	PIN_F1	2
NUM_ONES[4]	Output	PIN_F2	2
NUM_ONES[3]	Output	PIN_H1	2
NUM_ONES[2]	Output	PIN_H2	2
NUM_ONES[1]	Output	PIN_J1	2
NUM_ONES[0]	Output	PIN_J2	2
NUM_TENS[6]	Output	PIN_D1	2
NUM_TENS[5]	Output	PIN_D2	2
NUM_TENS[4]	Output	PIN_G3	2
NUM_TENS[3]	Output	PIN_H4	2
NUM_TENS[2]	Output	PIN_H5	2
NUM_TENS[1]	Output	PIN_H6	2
NUM_TENS[0]	Output	PIN_E1	2
NUM_THOUS[6]	Output	PIN_D4	2
NUM_THOUS[5]	Output	PIN_F3	2
NUM_THOUS[4]	Output	PIN_L8	2
NUM_THOUS[3]	Output	PIN_J4	2
NUM_THOUS[2]	Output	PIN_D6	2
NUM_THOUS[1]	Output	PIN_D5	2
NUM_THOUS[0]	Output	PIN_F4	2
RST	Input	PIN_R22	6
SER_RX	Input	PIN_B14	4
SER_TX	Output	PIN_A14	4

חזור לתוכן עניינים

SRAM_ADDR[17]	Output	PIN_Y5	8
SRAM_ADDR[16]	Output	PIN_Y6	8
SRAM_ADDR[15]	Output	PIN_T7	8
SRAM_ADDR[14]	Output	PIN_R10	8
SRAM_ADDR[13]	Output	PIN_U10	8
SRAM_ADDR[12]	Output	PIN_Y10	8
SRAM_ADDR[11]	Output	PIN_T11	8
SRAM_ADDR[10]	Output	PIN_R11	8
SRAM_ADDR[9]	Output	PIN_W11	8
SRAM_ADDR[8]	Output	PIN_V11	8
SRAM_ADDR[7]	Output	PIN_AB11	8
SRAM_ADDR[6]	Output	PIN_AA11	8
SRAM_ADDR[5]	Output	PIN_AB10	8
SRAM_ADDR[4]	Output	PIN_AA5	8
SRAM_ADDR[3]	Output	PIN_AB4	8
SRAM_ADDR[2]	Output	PIN_AA4	8
SRAM_ADDR[1]	Output	PIN_AB3	8
SRAM_ADDR[0]	Output	PIN_AA3	8
SRAM_CE_N	Output	PIN_AB5	8
SRAM_DQ[15]	Bidir	PIN_U8	8
SRAM_DQ[14]	Bidir	PIN_V8	8
SRAM_DQ[13]	Bidir	PIN_W8	8
SRAM_DQ[12]	Bidir	PIN_R9	8
SRAM_DQ[11]	Bidir	PIN_U9	8
SRAM_DQ[10]	Bidir	PIN_V9	8
SRAM_DQ[9]	Bidir	PIN_W9	8
SRAM_DQ[8]	Bidir	PIN_Y9	8
SRAM_DQ[7]	Bidir	PIN_AB9	8
SRAM_DQ[6]	Bidir	PIN_AA9	8
SRAM_DQ[5]	Bidir	PIN_AB8	8
SRAM_DQ[4]	Bidir	PIN_AA8	8
SRAM_DQ[3]	Bidir	PIN_AB7	8
SRAM_DQ[2]	Bidir	PIN_AA7	8
SRAM_DQ[1]	Bidir	PIN_AB6	8
SRAM_DQ[0]	Bidir	PIN_AA6	8
SRAM_LB_N	Output	PIN_Y7	8
SRAM_OE_N	Output	PIN_T8	8
SRAM_UB_N	Output	PIN_W7	8
SRAM_WE_N	Output	PIN_AA10	8

חזור לתוכן עניינים

סיכום ומסקנות

במהלך עבודה זו הייתה לנו ההזמנות להביא לידי ביטוי כל מיני כלים ושיטות שלמדנו במהלך הסמסטר, תוך מתן דגש על התכנון היררכי.

בנוסף, העבודה תרמה לנו בהבנת השימוש בשפת VHDL ושיפרה רבות במיומנות כתיבת קוד לתיאור חומרה ודיבוגו.

והאמת הגענו לקורס עם קיבעון מחשבתי שכנראה נוצר לנו בקורסים קודמים כגון: תכן לוגי ושפת C. קיבעון מחשבתי זה בא לידי ביטוי בכך שחלק ניכר שמהלך ביצוע העבודות שניתנו לאורך הסמסטר "חשבנו כיצד היינו מתכננים רכיב כלשהו בחומרה כמו שהיינו עושים בתכן לוגי" ועבודה זאת שיחררה קיבעון זה, והתכנון התחיל לזרום בצורה יותר אינטואטיבית תוך כדי ביטחון מתגבר בשינויים בקוד.

כמו כן, עבודה זו פתחה לנו אשנב לעולם התקשורת טורית/מקבילית. נחשפנו למהירות המטורפת שהמידע עובר. והיה בעבורנו אתגר לא פשוט להתגבר על האופן שבו נדרשנו ליצור את הממשק בין כל הרכיבים.

בנוסף שמנו לב שהיה עלינו לבצע המון דיבוגים למערכת, והיו המון גרסאות עד שהצלחנו להגיע לגרסא הסופית. ואם נדמיין לרגע שכל בדיקה כזאת לצורך העניין הייתה צריכה להתבצע עם מימוש , הדבר היה דורש המון המון כסף ומשאבים, ופשוט השיטה הזאת של שימוש ברכיבים מתוכנתים ,FPGA, חוסכת המון ! ועבודה זאת תרמה לנו בהבנה כמה השיטה הזו תורמת להנדסה!

פרח או ציור יפה בסוף

לבקשתך 😊