

המחלקה להנדסת חשמל ואלקטרוניקה

Token Ring 802.5 – תכן לוגי

: מוגש עייי

ת.ז 208776328	ן דרויש	ערין
---------------	---------	------

208778324 גואד שעיב ת.ז

מחמוד חגה ת.ז 318396355

בהנחיית: מר משה שדה.

תוכן עניינים

מבוא	. 1
4תיאור המערכת	.2
4 2.1	
4 2.2 דוגמא להרצה	
6 ברישות המערכת 2.3	
6 מלבני של המערכת	
6 תרשים מלבני •	
7 • מפרט טכני	
תכן המערכת	.3
8 פעולת המערכת 3.1	
10 סכמות חשמליות	
11. Shift Register •	
12•	
13	
14	
15STOREDROM מעגל	
71PERMISSION 1 מעגל •	
19ADDRESS מעגל	
21readx1 מעגל •	
22writex1 מעגל •	
23 counter256 מעגל	
24 HELLO מעגל	
סימוכיו	.4

: מבוא

רשתות טבעות אסימון- Token ring עושות שימוש באסימון בגודל 24 סיביות המועבר בין המחשבים ברשת, כאשר רק מחשב שברשותו האסימון יכול לשדר. האסימון מועבר בין המחשבים בסדר קבוע. הסכם זה גורם לכך שאין התנגשויות ברשת, מצד שני עלולה להגרם הרעבה של תחנות (עלול להגרם מצב בו תחנה שצריכה לשדר לא מקבלת אסימון ולכן לא משדרת).

תקן ה־ Token ring נוצר על ידי אולוף סודרבלום (Olof Söderblom) בשנות ה-60 המאוחרות, בשנת 1970 הוא אומץ על ידי חברת IBM שהכניסה אותו לשימוש נרחב עד אמצע שנות ה-80. בשנת Token ring בשנת ידי ה־ IEEE 802.5 עם מספר מאוחר יותר ה־ IBM הנוגעים בעיקר לסוג

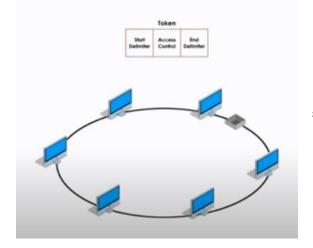
הכבילה ולטופולוגיה הפיזית של

הרשת.

ברשת טבעות אסימון, כל המארחים / תחנות העבודה מחוברים בטופולוגיית הטבעות .האסימון, מסגרת ריקה, מופץ ברציפות על הטבעת. אסימון ריק כולל שלושה שדות בלבד :



- .2 בקרת גישה.
- .3 תוחם מסגרת סוף.



האסימון הוא כמו מונית, שעוברת על לולאה בשכונה .אם לתחנת עבודה אין מה להעביר, היא מעבירה את האסימון .אם יש לה משהו להעביר, היא תופסת את האסימון ומצרפת אליו נתונים ושולחת אותם חזרה לטבעת.

2. תיאור המערכת:

2.1 פעולת המערכת:

- המשתמש יטען נתונים דרך המקלדת. 💠
- 💠 המשתמש יבחר כתובת לשלוח הנתונים אליה.
- המערכת מקבלת את הנתונים אחרי לחיצת המשתמש על כפתור ישליחהיי, כאשר לייתוקן נמצא אצל המשתמש.
 - הנתונים יעברו לכתובת שנשלה.

: דוגמא להרצה 2.2

בואו להשתמש בדוגמה כדי להדגים איך זה עובד.

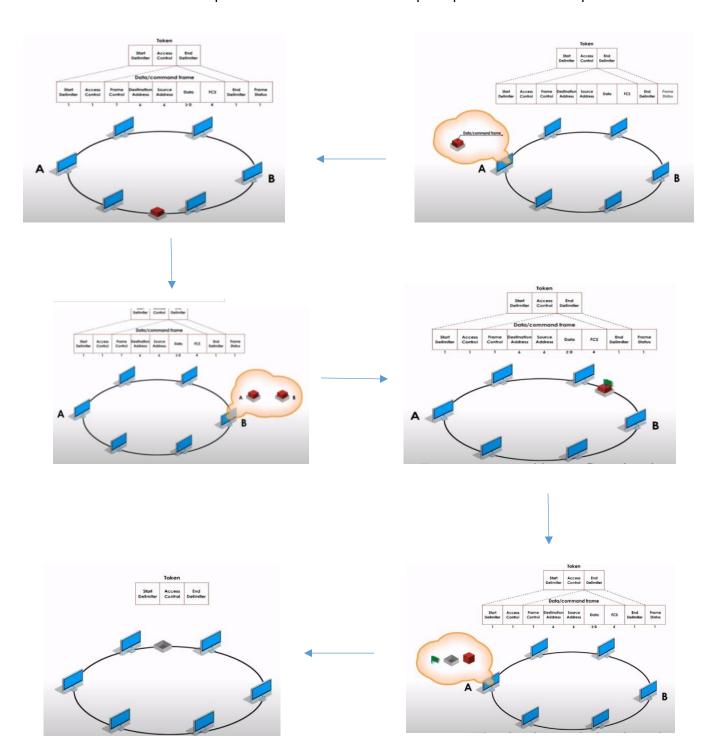
B. יש עיסוי שישלח לתחנת העבודה A נניח שלתחנת העבודה

הוא תופס את האסימון החולף ומצרף את הנתונים שלו ומוסיף מידע כותרת, כגון בקרת מסגרת, כתובת יעד, כתובת מקור, FCS ומצב מסגרת. כעת האסימון הריק הופך למסגרת נתונים / פקודה.

האסימון משוחרר חזרה לטבעת, ואז נבדק על ידי כל תחנת עבודה עוקבת, אשר בודקים את כתובת היעד מול כתובתם שלהם, ואז מעבירים אותה אם הם אינם תואמים .כאשר מסגרת הנתונים מגיעה ליעד, תחנת העבודה B מעתיקה את המסגרת, משנה את מצב המסגרת ואז משחררת את האסימון לרשת. מצב המסגרת מספק אישור לכך שההודעה הועברה בהצלחה . זה כמו להניף דגל, לומר לאחרים : הנתונים כבר מועברים ומועתקים. כאשר האסימון חוזר לשולח, תחנת העבודה A מסירה את ההודעה מהאסימון ואז משחררת אסימון ריק חדש בחזרה לרשת. האסימון הופך להיות זמין שוב, מוכן לקבל את ההודעה הבאה. כאשר תחנת עבודה רוצה להעביר נתונים, זה צריך להחזיק את האסימון עד שהוא נעשה. אם מארחים אחרים רוצים לשלוח נתונים עליהם להמתין.

המכללה האקדמית להנדסה אורט בראודה

המחלקה להנדסת חשמל ואלקטרוניקה

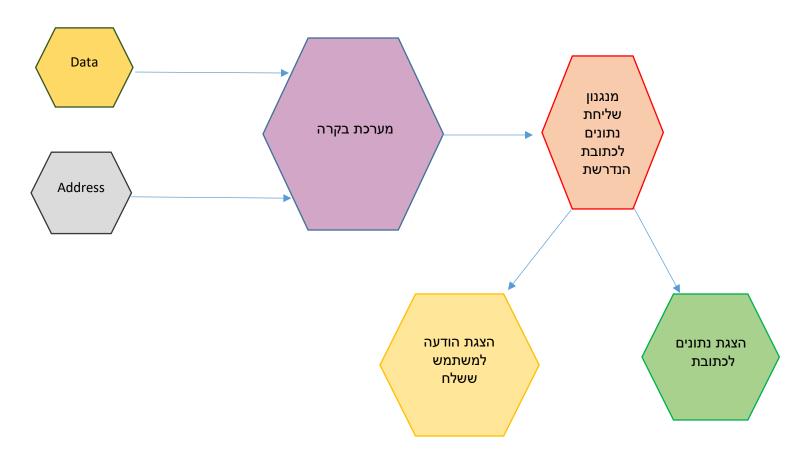


2.3 דרישות המערכת/ מפרט פונקציונלי:

- המערכת עובדת בטופולוגית כוכב, אשר כל רכיב מחובר לזה שלפניו ולזה שאחריו בלבד.
 - לוגי בין הרכיבים- רק רכיב אשר מחזיק ב Token המערכת משתמשת בהעברת Token לוגי בין הרכיבים- רק רכיב אשר מחזיק ב Token
- . המערכת גם מוגדרת להעברת/מסירת מידע אך ורק לפי סדר העדיפות של התחנות.

2.4 תרשים מלבני של עבודת המערכת:

• תרשים מלבני:



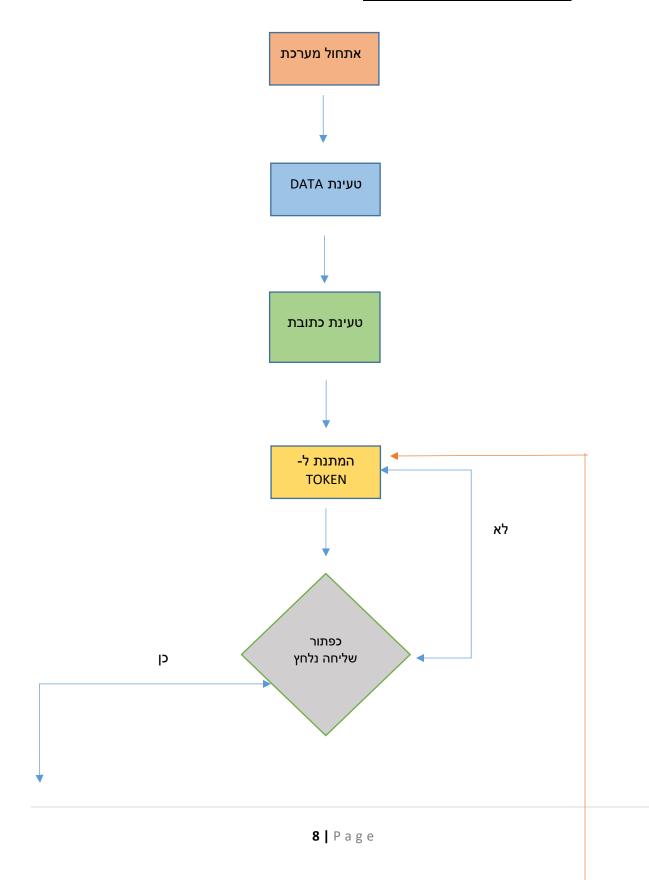
<u>:מפרט טכני</u>

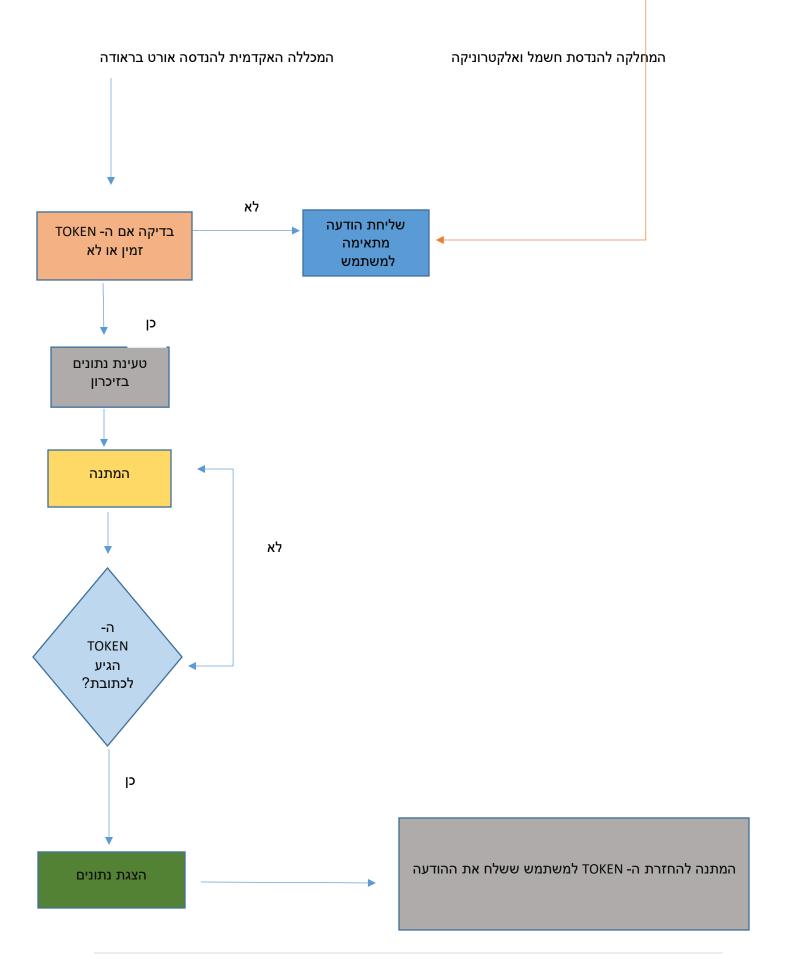
	÷		
_		ı	
	1	L	
	7	г	١

פרמטרים	רכיב
Tpcq=13[ns]	D-FF
Tccq=5[ns]	
Tsu=20[ns]	
Th=5[ns]	
Tpd=14[ns]	Counter
Tcd=14[ns]	
Tsu=11[ns]	
Th=2[ns]	
Tpd=20[ns]	Mux
Tcd=13[ns]	
Tpd(clk->out)=25[ns]	Register
Tpd(MR'->out)=30[ns]	
Tcd(clk->out)=20[ns]	
Tcd(MR'->out)=13[ns]	
Tsu=13[ns]	
Th=5[ns]	
Tpd=4.9[ns]	NOT-Gate
Tcd=2.5[ns]	
Tpd=14[ns]	AND-Gate
Tcd=7[ns]	
Tpd=22[ns]	OR- Gate
Tcd=14[ns]	
Tpd=12[ns]	4-Input And
Tcd=7[ns]	
Tpd=12[ns]	4 Input-OR
Tcd=8[ns]	
Tsu=2[μs]	ROM
Tpd=45[ns]	
Th=o[ns]	RAM
Tpd=35[ns]	Shift Register
Tcd=8[ns]	
Tsu=10[ns] Th=0[ns]	

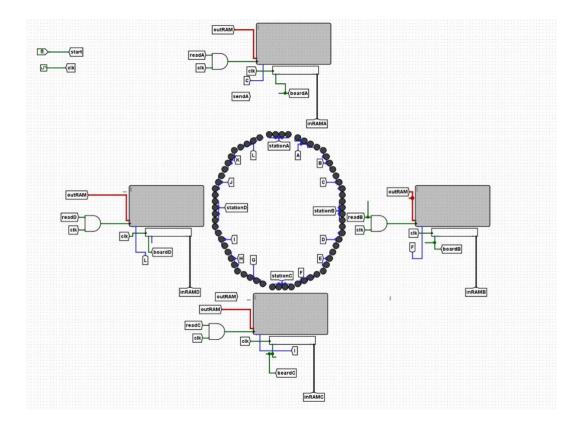
<u>מכן המערכת:</u> 3

3.1 אלגוריתם פעולת המערכת:





: סכמות חשמליות 3.2



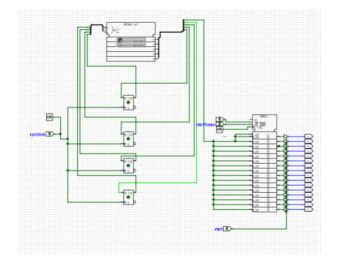
: קלט כללי למערכת

, Token לקבלת להעברת Token עם מידע כלשהו להעברת Token עם מידע להעברת בחירת Station בחירת הפעלה . start : כפתור ההפעלה

פלט כללי למערכת:

תצוגת לד ראשי בצבע כחול בזמן קבלת ה Token בתחנה המתאימה ותצוגת לדים משנית כל עוד ה Token עובר בטבעת.

Shift register:



. <u>קלט:</u> שעון

פלט: עלייה אחת כל 16 עליות לשעון הנקלט

<u>תיאור :</u>המעגל נותן אחד עליית שעון כל 16 עליות מהשעון הנקלט

מסלול קריטי אשר קובע את תדר השעון במעגל:

 $T(in) \rightarrow D-FF : Tsu(in) >= Tsu(D-FF) + Th(D-FF) = 20ns + 5ns = 25[ns].$

Tclk >= Tsu(D-FF)=20[ns].

D-FF \rightarrow ROM : Tclk>= Tpcq(D-FF)+Tsu(ROM)=13ns+2 μ s=2.013[μ s].

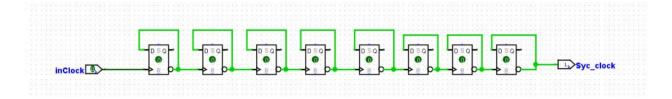
 $\label{eq:ROM} \textbf{ROM} \rightarrow \textbf{SRG: Tclk} >= \textbf{Tpd(ROM)} + \textbf{Tsu(SRG)} = \textbf{45ns} + \textbf{10ns} = \textbf{55[ns]}.$

Tcd(ROM) >= Th(SRG) : 7[ns] >= 0[ns]

 $SRG \rightarrow OUT: Tclk >= Tpd (SRG) = 35[ns]$

Tclk = 20[ns]

מחלק תדר:



תיאור: מעגל זה נועד להכפיל את זמן מחזור השעון פי 256 כאשר במעגל שלנו נשתמש באות המוכפל בחלק מתתי המעגלים על מנת לתזמן השהיות נדרשות לשם פעולת המעגל.בעבור המוכפל בחלק מתתי המעגלים על מנת לתזמן השהיות נדרשות לשם פעולת המעגל. החיבור הנייל ביציאה Q של כל D-FF יתקבל אות שמחזורו יהיה פי 256 ממחזור אות השעון.

$$Tout = 2*2*2*2*2*2*2*2*Tin = 256Tin$$

CLK_IN (אות השעון המקורי) אות השעגל:

CLK_OUT (אות השעון המוכפל: (אות השעון המוכפל)

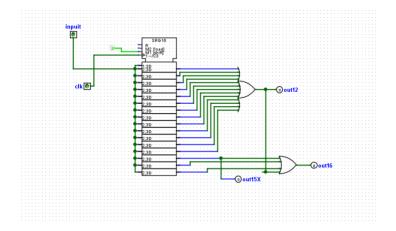
ניתוח זמנים:

$$Ts(in) \ge Tsu(D-FF) + Th(D-FF) = 20+5 = 25[ns]$$

$$Tclk \ge 5*Tpcq(D-FF)+Tsu(D-FF) = 8*13+20 = 124[ns]$$

$$Tccq(D \neg FF) \ge Th(D \neg FF) \longrightarrow 5[ns] \ge 5[ns]$$

SHIFT:



תיאור במעגלים על מנת המעגלים ב16 בחלק מתתי המעגלים על מנת לתזמן השהיות בדרשות לשם פעולת המעגל. נדרשות לשם פעולת המעגל.

CLK ,INPUT <u>- קלט</u>

OUT 16,OUT 12, <u>פלט:</u>

חישוב זמנים:

 $IN \rightarrow Shift$:

 $Ts(in) \ge Tsu(shift-reg) + Th(shift-reg)=10ns$

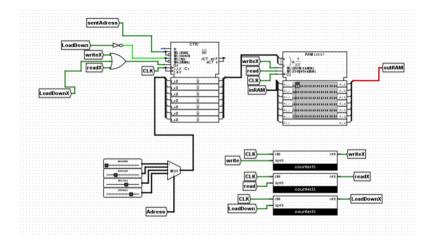
Shift→OR:

Tclk≥ Tpd(shift-reg)+tTsu(OR)=35ns

OR→ OUT:

 $Tclk \ge Tpd(or) = 22[ns]$

RAM:



תיאור: קריאת הנתונים הנדרשים להעביר אותם בין התחנות, כמו Token או חבילות שהקצאנו אות כ 256 ביט.

קלט: גישה לכתיבה, לקריאה, לאיפוס המונה של הראם.

. Ascii ב Token <u>פלט:</u> פלט של ח

ניתוח זמנים:

In \rightarrow Counter: Tsu(in) \geq Tsu(counter)+Thold(counter)=11ns+2ns=13 [ns].

Counter \rightarrow OR: Tclk ≥ Tpd(counter)=14 [ns].

In \rightarrow NOT: Ts(in) \geq Tpd(not) = 4.9 [ns].

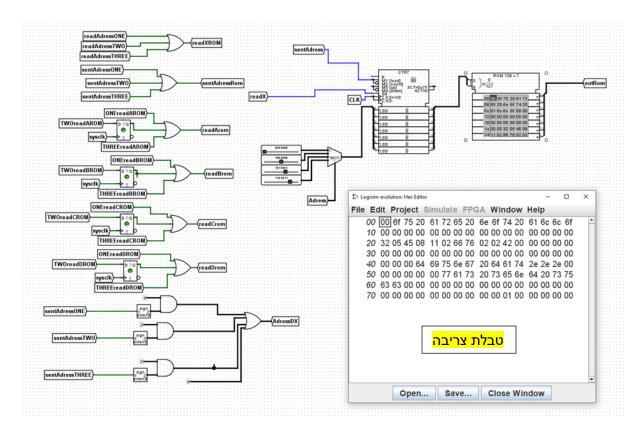
MUX → Counter: $Tcd(MUX) \ge Th(counter) : 7ns>=2[ns].$

Counter → RAM : Tcd(counter) ≥Th(RAM) : 14ns≥0 [ns]

OR→ Counter:

Tclk≥ Tpd(OR)+Tsu(Counter)=22ns+11ns=33[ns].

מעגל STORED ROM:



: תיאור מעגל

המעגל פעול כאשר אחד מהמשתמשים מנסה לשולח נתונים אל הזיכרון, המעגל שולח למסך המשתמש הודעה מתאימה אם הוא יכול לשלוח נתונים או לא, וגם אם הנתונים שלו נשלחו בהצלחה .

clk,sentaddress,address,read: קלט

outROM,readx: פלט

: חישוב זמנים

In→ counter:

 $Tsin \ge Tpd(mux) + th(counter) + tsu(counter) = 17ns + 2ns + 11ns = 30ns$

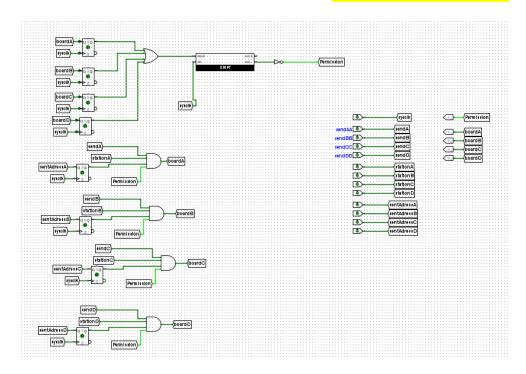
Counter→ROM:

Tclk≥TPd(counter)+Tsu(rom)=14ns+2μs=2.014μs

Rom→out:

Tclk≥Tpd(rom)=45 ns

:PERMISSION1 מעגל



תיאור להמשיך ולהמשיך להעביר את לקבל את ה לקבל הוא לאפשר להעביר את תפקיד המעגל הוא לאפשר אם לקבל לתחנה הבאה token לתחנה הבאה .

sysclk,sendA,sendB,sendC,send,stationA,stationB,stationC,stationD, $\underline{\cdot}$ gddressA,addressB,AddressC,AddressD,sent(A,B,C,D).

PERMISSION <u>: פלט</u>

<u>חישוב זמנים :</u>

In→ FF1:

 $T\sin \ge T\sin(FF1) + Th(FF1) = 20ns + 5ns = 25[ns].$

 $FF1 \rightarrow AND$:

 $Tclk \ge Tpcq(FF1) + Tsu(AND) = 13[ns].$

AND→FF2:

 $Tclk \ge Tpd(AND) + Tsu(FF2) = 14 + 20 = 34[ns].$

FF2→shift:

 $Tclk \ge Tpcq(FF2) + Tsu(shift) = 13[ns].$

 $FF2 \rightarrow OR$:

 $Tclk \ge TpcQ(FF2) + Tsu(OR) = 13[ns].$

OR→Shift:

 $Tclk \ge Tpd(OR) + Tsu(shift) = 22ns + 300ns = 322[ns].$

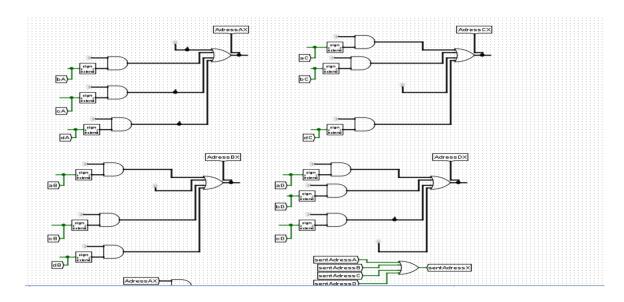
Shift \rightarrow NOT:

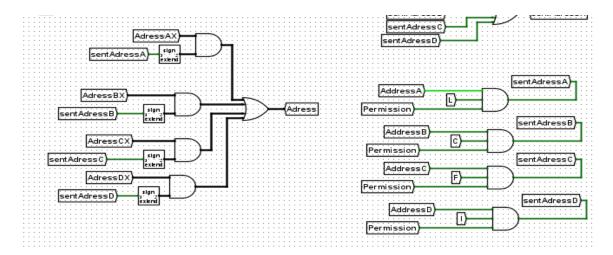
 $Tclk \ge Tpd(shift) + Tsu(NOT) = 195[ns].$

NOT \rightarrow out:

Tclk≥ Tpd(NOT)=4.9[ns].

: ADDRESS מעגל





תיאור: תפקיד המעגל, הוא לספק ל ram את הכתובת הדרושה לשליחה וקבלת חבילות token . ומספק לנו איזו תחנה שלחה את ה-

sentaddressA,sentaddressB,sentaddressC,sentaddressD $\underline{\cdot}$ addressX,addressB,addressC,addressD,addressA $\underline{\cdot}$ פלט

חישוב זמנים:

In→ AND:

 $Tsin \ge Tsu(AND) + Th(AND) = 0[ns].$

AND \rightarrow OR :

 $Tclk \ge Tpd(AND) + Tsu(OR) = 14[ns].$

 $OR \rightarrow AND$:

 $Tclk \ge Tpd(OR) + Tsu(AND) = 22[ns].$

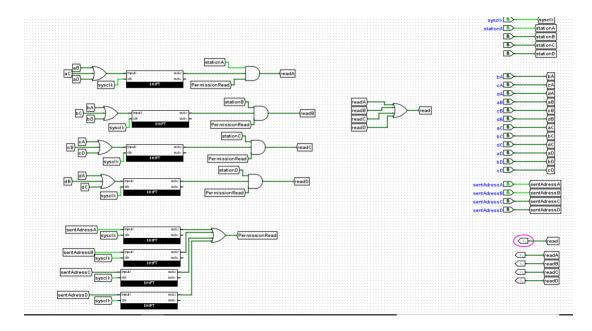
AND \rightarrow OR:

 $Tclk \ge Tpd(AND) + Tsu(OR) = 14[ns].$

 $OR \rightarrow OUT$:

 $Tclk \ge Tpd(OR) = 22[ns]$.

:readx1 מעגל



תיאור: תפקיד המעגל הוא לאפשר קריאה מ-הראם אחרי קבלת כתובת היעד המתאימה מהשולח .

read <u>: פלט</u>

: חישוב זמנים

In→shift :

 $Tsin \ge Tpd(OR) + Tsu(shift) = 22ns + 300ns = 322[ns].$

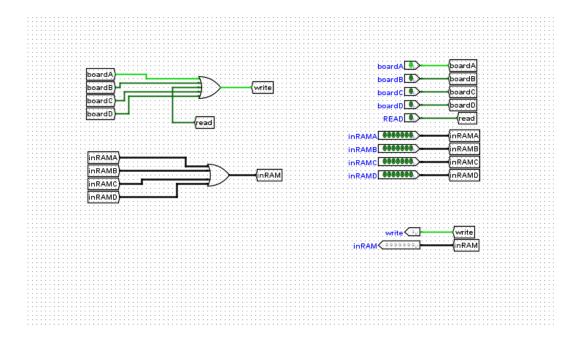
Shift→ AND:

 $Tclk \ge Tpd(shift) + Tsu(aAND) = Tpd(shift) = 195[ns].$

AND→OUT :

Tclk≥ Tpd(AND)=14[ns].

: writex1 מעגל



תיאור: תפקיד המעגל הוא לאפשר לתחנה לכתיבת נתונים אל תוך ה-ראם , אחרי שנקרא ממנו .

borad(A-D),read,inram(A-D) : קלט

write ,inram <u>: פלט</u>

חישוב זמנים:

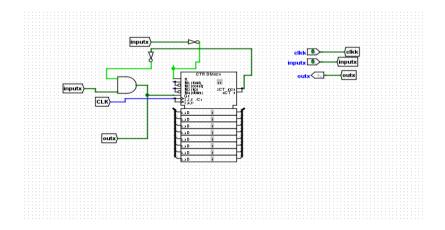
 $In \rightarrow OR$:

 $Tsin \ge Tsu(OR) + Th(OR) = 0[ns].$

OR**→**OUT:

 $Tclk \ge Tpd(OR) = 22[ns].$

:counter256 מעגל



<u>תיאור:</u> מכיוון שהגדרנו שאורך המידע הוא כ256ביט , אז משתמשים במונה להגביל את כמות הביטים של המידע בראם , זאת אומרת אחרי שהגענו ל256 ביט המונה מתאפס ועוצרים את הכתיבה ב-ראם.

clk,inputx : קלט

outx <u>: פלט</u>

חישוב זמנים:

In→ Counter:

 $Tsin \ge Tpd(AND) + Tsu(Counter) + Th(Counter) = 14 + 11 + 2 = 27[ns].$

Counter→NOT:

 $Tclk \ge Tpd(Counter) + Tsu(NOT) = 14[ns]$.

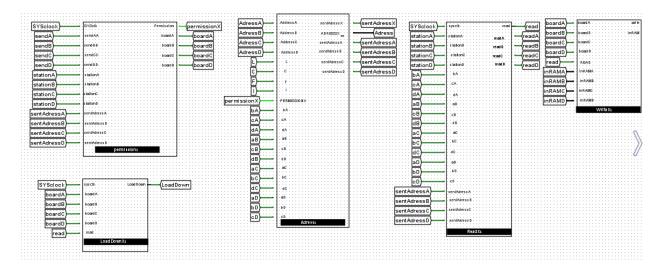
NOT \rightarrow AND :

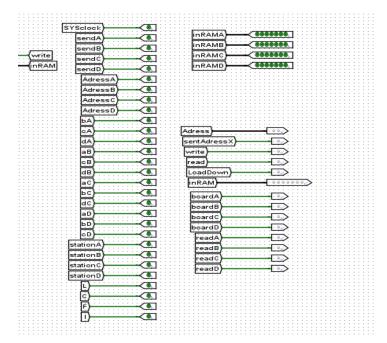
 $Tclk \geq Tpd(NOT) + Tsu(AND) = 4.9[ns].$

AND \rightarrow OUT:

Tclk≥ Tpd (And)=14[ns].

:HELLO מעגל





תיאור: סכמה כללית של המעגלים שבנינו.

 $4.96 \mathrm{Mhz}$ ולכן תדר השעון 2.014 $\mu \mathrm{s}$ זמן המחזור הקריטי ביותר הוא

4. <u>סימוכין :</u>

דפי נתונים: 👃

RAM RAM

D-FF <u>74SL74</u>

AND-GATE 74LS08

4-INPUTS-AND-GATE 74HC21

OR-GATE 74LS32

4-INPUTS-OR-GATE 74HC4072

NOT-GATE 74LVC2G04

COUNTER 74HC590

MUX 74HCT151

SHIFT-REGISTER <u>HEF4557B</u>

EEPROM AT27C516

REGISTER SN74LVC32373A

מקורות מידע: 🚣

- https://www.ionos.com/digitalguide/server/know-how/token-ring/
- https://www.digitalwhisper.co.il/files/Zines/0x6C/DW108-3-TokenRingFDDI.pdf