

תוכן העניינים:

2	פרק 10
2	אוגרים ומונים
2	אוגרים (Registers):
2	סיכום כללי:
9	שאלות:
11	תשובות סופיות:
12	מונים (Counters):
12	סיכום כללי:
18	שאלות:
19	תשובות סופיות:

פרק 10

אוגרים ומונים

אוגרים (Registers):

סיכום כללי:

מעגלים סדרתיים רבים הינם בעלי שימוש רחב בתור MSI (medium scale Integration) ו-LSI (Large Scale Integration) של שבבי IC (Integration Circuits). נעסוק בשני סוגים מרכזיים הנקראים **אוגרים (Registers)** ו**מונים (Counters)**.

אוגרים נחלקים לשני סוגים מרכזיים:

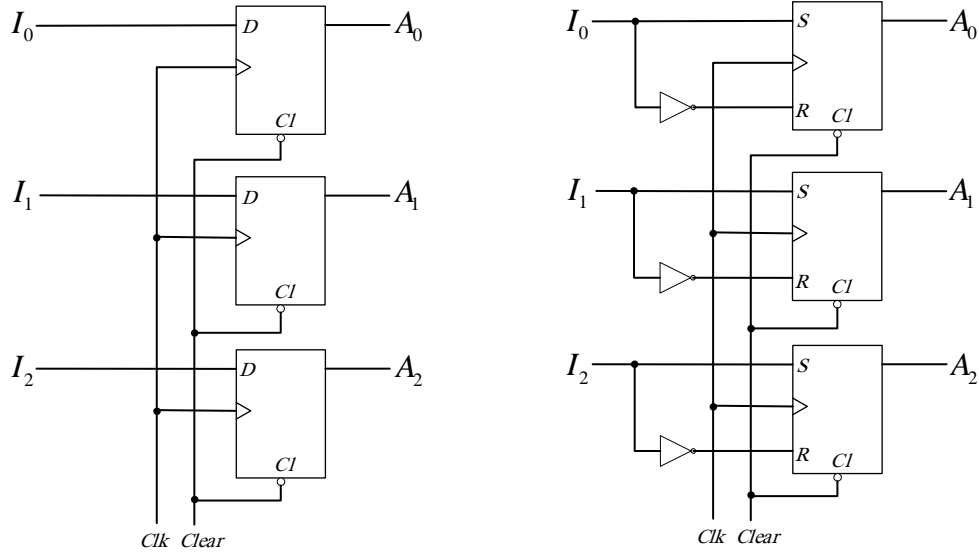
1. אוגר מקבילי (Parallel Register) המשמש בעיקר לאחסון מידע.
2. אוגר הזזה (Shift Register) המשמש בעיקר למגוון פעולות עיבוד של מידע.

אוגר מקבילי (Parallel Register):

אוגר מורכב מ-**תאים (cells)** אשר ממומשים באמצעות רכיבי זיכרון (Flip Flops). מספר תאי הזיכרון מגדיר את **גודל האוגר (Length)**. אוגר באורך n יכול לאחסן 2^n ערכים שונים (Binary combinations/values).

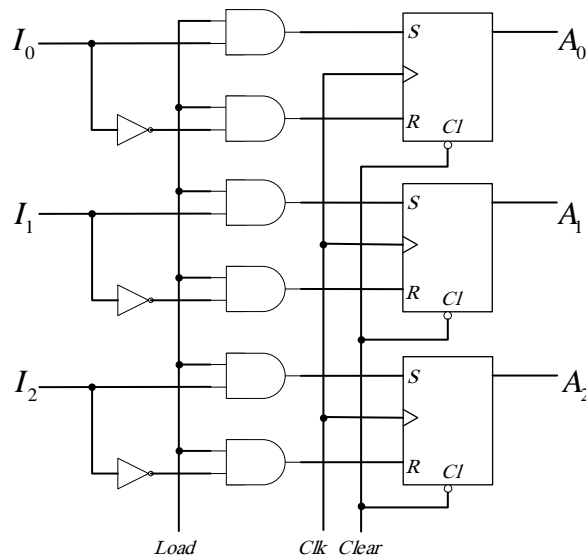
מבנה יסודי של אוגר מקבילי:

להלן שתי תצורות של אוגר מקבילי בגודל 3 יחידות.



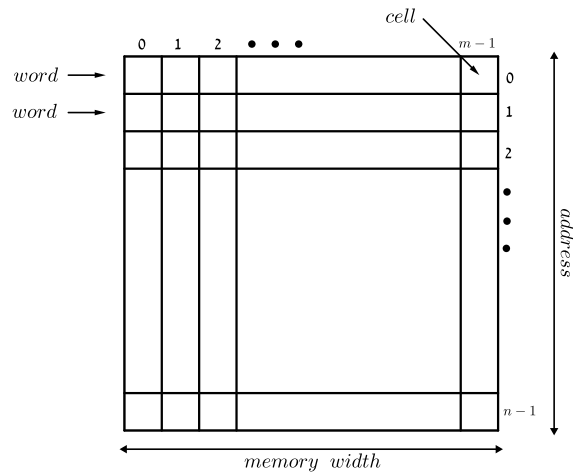
מבנה מוכלל של אוגר מקבילי:

כדי לשלוט על ערכי האוגר מגדירים כניסת LOAD שתפקידה לשלוט על כניסת המידע לתוך האוגר ולמנוע מצב בו פולסי השעון המחזוריים ישלטו במוצאיו.



מבנה של יחידת זיכרון והגדרות:

- יחידת זכרון מורכבת מאוסף של אוגרים שכל מטרתם היא לאגור מידע בהתאם לצרכי המערכת.
- את המידע מארגנים ביחידות הנקראות **מילים** (words) כאשר כל מילה נחשבת בעלת ערך בפני עצמה.
- אורך מילה נקרא **רוחב הזיכרון** (memory width) ומסומן ב- m .
- כדי לפנות בצורה חד-חד ערכית לכל אוגר מקובל לתת להם **כתובות** (addresses). כך לאוגר הראשון תינתן הכתובת 0 ולאוגר האחרון תינתן הכתובת $n-1$.
- הקיבולת (capacity) מוגדרת בתור כמות הביטים הכללית שאוגר יכול להכיל. יחידת זכרון עם n אוגרים שמאחסנים מילים באורך m היא בעלת קיבולת mn .



דוגמאות לקיבולת של יחידות זכרון:

מקובל לייחס למילה אורך של 8 ביטים והיא נקראת **בית** (Byte), אולם במערכות מודרניות מילים מכילות גם 32 ביטים ו-64 ביטים. (למשל: מערכות הפעלה הפועלות על 32 ביט או 64 ביט). הקיבולת של יחידות זיכרון גדולות נכתבות באמצעות:

$$k = 2^{10} \quad (\text{kilo})$$

$$M = 2^{20} \quad (\text{Mega})$$

$$G = 2^{30} \quad (\text{Giga})$$

$$T = 2^{40} \quad (\text{Tera})$$

קריאה וכתובה מאוגר מקבילי:

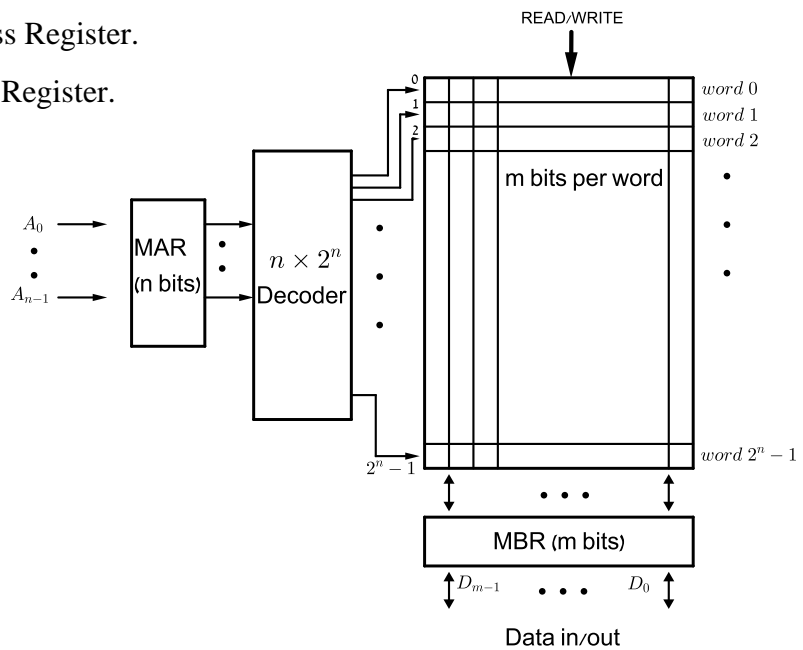
בשונה מיחידת זיכרון ROM שמבוססת על מערכים מסוג PLA או PAL, יחידת זיכרון המורכבת מרכיבי זיכרון (FF) מאפשרת פעולות של כתיבת מידע לתוכה וקריאה של מידע ממנה.

- לתהליך אחסון מידע קוראים **כתיבה** (writing) והוא תהליך **הרסני** (destructive).
- לתהליך לקיחת המידע קוראים **קריאה** (reading) והוא אינו תהליך הרסני.
- לרכיב מסוג זה קוראים RWM – Read/Write Memory.
- כאשר תהליכי הקריאה והכתיבה מתבצעים בפרקי זמן זהים, מקובל לקרוא לרכיב שכזה בשם RAM – Random Access Memory (או בשמו המלא: Random Access RWM).
- יחידת זכרון RAM היא volatile בעוד שמערכי ROM הם nonvolatile.

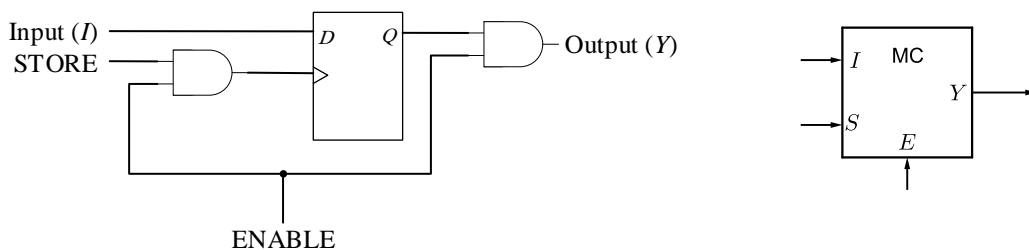
מבנה כללי של RAM:

MAR – Memory Address Register.

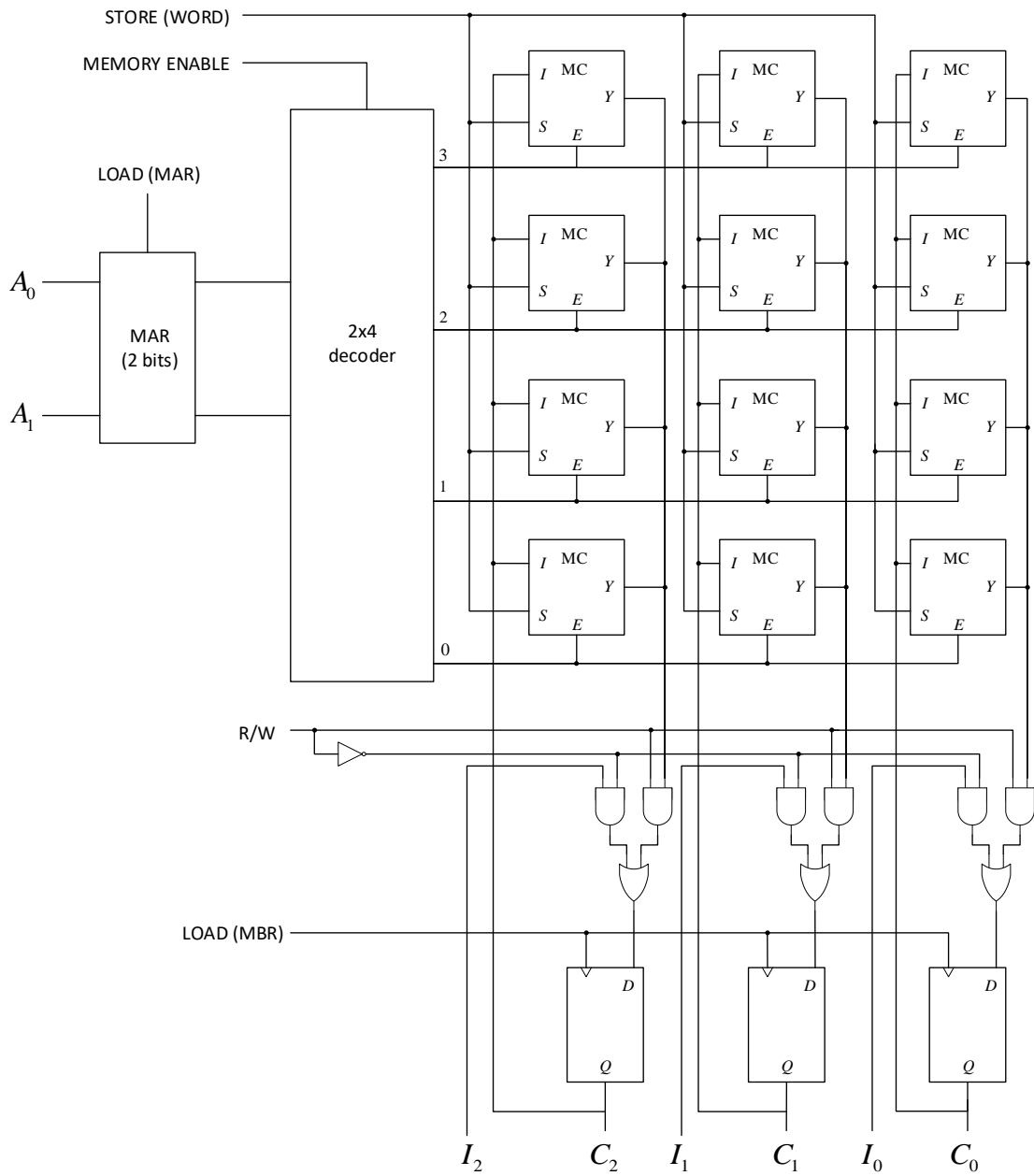
MBR – Memory Buffer Register.



מבנה תא זיכרון בודד (Memory Cell - MC):

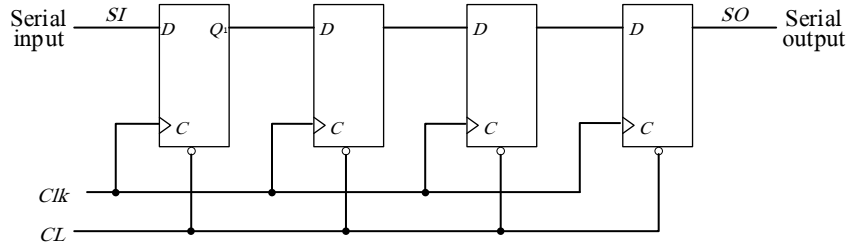


מבנה כללי של RAM בגודל 2×3 :



אוגר הזזה:

אוגר שבו רכיבי הזיכרון משורשרים אחד לשני:



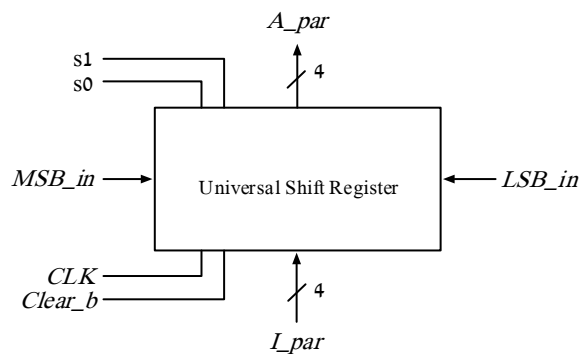
סוגי אוגרים:

אוגר הזזה שמעביר מידע מצד אחד לשני בלבד נקרא **אוגר חד כיווני (unidirectional)**.
 אוגר שמעביר מידע לשני הצדדים נקרא **אוגר דו-כיווני (bidirectional)**.
 קיימים 4 סוגים של אוגרים:

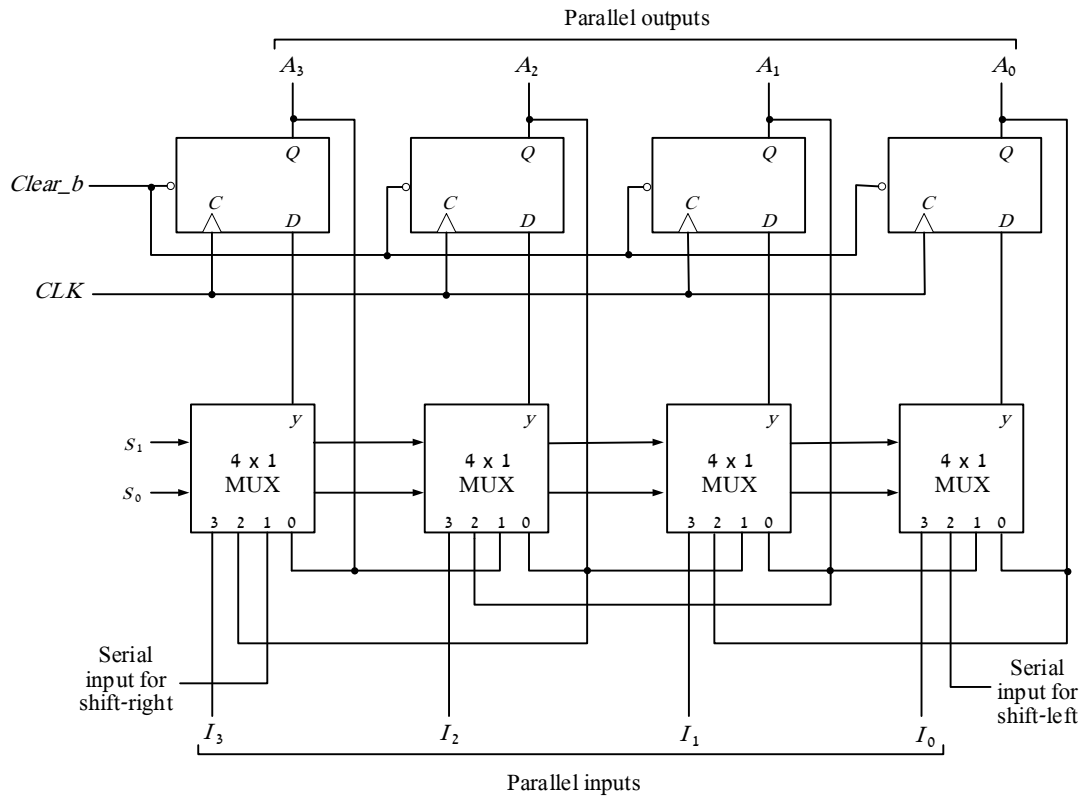
1. אוגר serial in / serial out (SISO).
 2. אוגר serial in / parallel out (SIPO).
 3. אוגר parallel in / serial out (PISO).
 4. אוגר parallel in / parallel out (PIPO).
- את כל 4 הסוגים ניתן לאחד למבנה אחד באמצעות שערים לוגיים.
 לאוגר מסוג זה קוראים **אוגר אוניברסלי (universal)**.

אוגר אוניברסלי:

תרשים כללי של אוגר אוניברסלי:



מבנה מפורט של אוגר אוניברסלי :



שאלות:

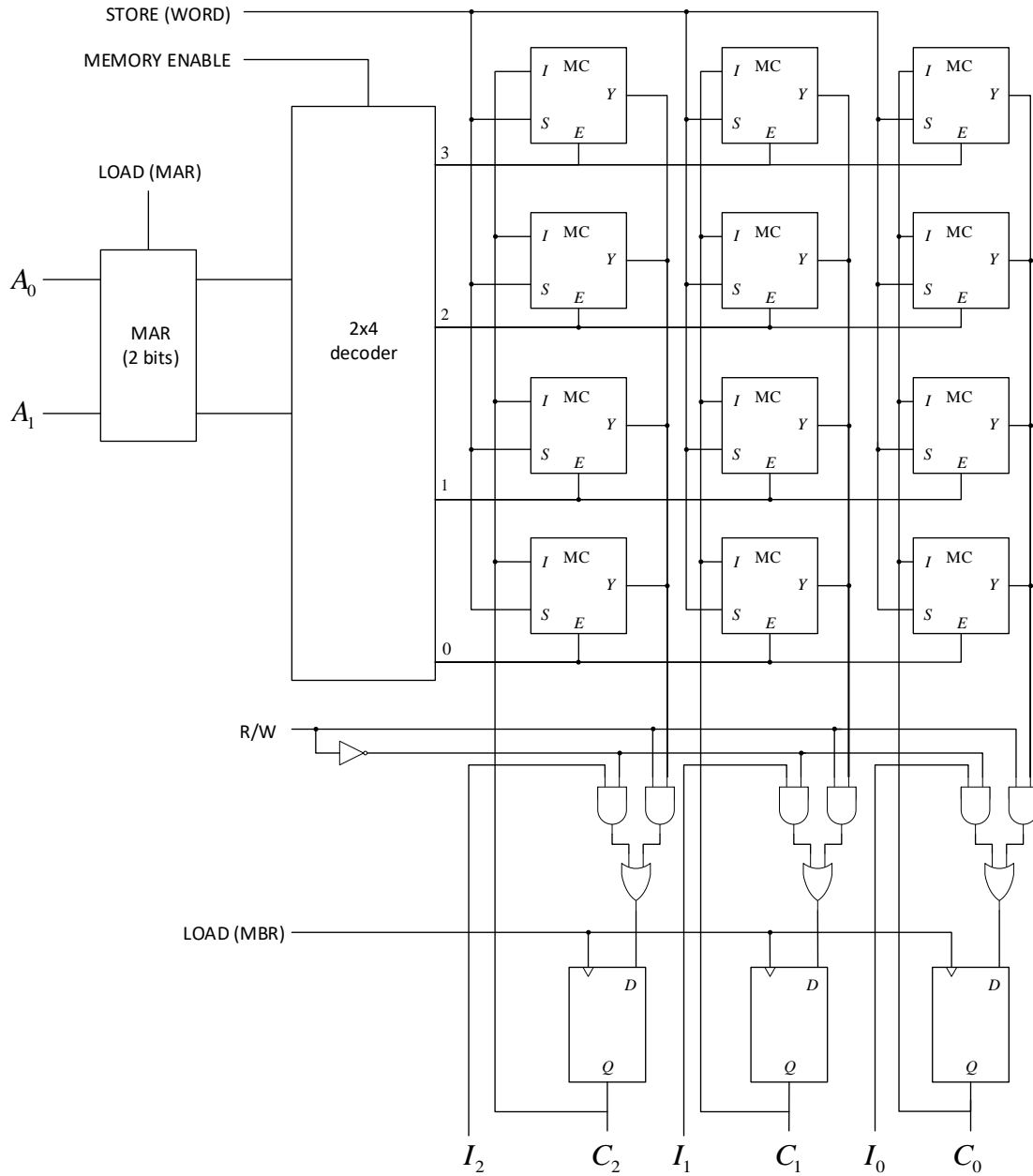
- (1) ממש אוגר מקבילי שמקבל מילה $I_3I_2I_1I_0$ ומכיל כניסת LOAD וכניסת clk באמצעות D-FF.
- (2) בשאלה זו נממש מסכם מקבילי של מילים בינאריות $A = a_3a_2a_1a_0$ ו- $B = b_3b_2b_1b_0$ באמצעות שני אוגרים מקבילים Reg1 ו-Reg2 ו-4 יחידות של FA.
- א. סרטט דיאגרמה לוגית של אוגר Reg1 בעלת כניסות: Load, Clear, Data_in, Data_out
- ממש את האוגר באמצעות רכיבי זיכרון מסוג D-FF והתייחס ל- Data_in ול-Data_out בתור מספרים בינארים בני 4 ביטים. היעזר בשערים לוגים במידת הצורך. האם יש צורך בשינוי הדיאגרמה עבור Reg2?
אם כן - מהו? אם לא - נמק.
- ב. חבר מעגל צירופי מתאים המורכב מ-4 יחידות של FA בין שני האוגרים על מנת לבצע פעולת חיבור.
שים לב לאופי החיבור:
- מתקבל מספר $A = a_3a_2a_1a_0$ אשר נכנס ל-Reg1.
 - מעבירים את A למעגל החיבור ומשם ל-Reg2.
 - אחריו מתקבל מספר $B = b_3b_2b_1b_0$ אשר נכנס ל-Reg1.
 - מתבצע חיבור באמצעות המעגל הצירופי.
 - התוצאה יוצאת מ-Reg2.
- על המעגל הכולל להכיל את הכניסות האסינכרוניות הבאות: Load1, Load2, Clear. (כלול גם את כניסת השעון CP).
- ג. סרטט דיאגרמת זמנים מתאימה לאופי פעולת הרכיב.

3) לפיך המעגל הבא.

תאר את השלבים שיש לבצע על מנת :

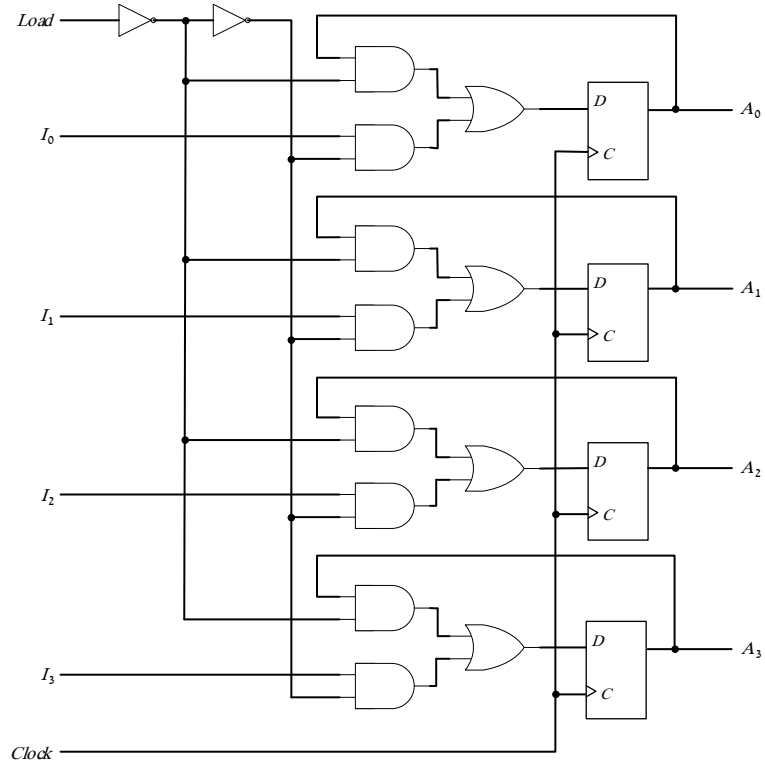
א. לכתוב את הערך 5 לאוגר שבכתובת 01 (למילה שכתובתה 01).

ב. לקרוא את התוכן של המילה הרביעית.



תשובות סופיות:

(1) להלן תרשים מתאים:



(2) ראה פתרון מלא בסרטון הוידאו.

(3) ראה סדר פעולות בסרטון הוידאו.

מונים (Counters):

סיכום כללי:

הגדרה:

אוגר שעובר דרך סדרת מצבים מוגדרת מראש נקרא **מונה** (Counter).

כניסות מונה:

למונה יתכנו כניסות סינכרוניות (כגון כניסת שעון) או כניסות אסינכרוניות (כגון ספירה של הפרעה מסוימת במערכת).

המונה הבינארי:

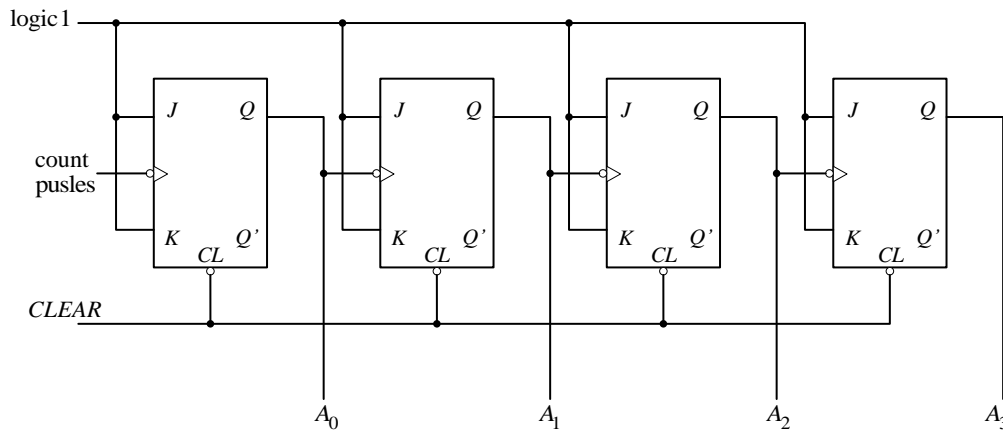
מונה שעובר אחר סדרת מצבים המתאימה לרצף בינארי נקרא מונה בינארי בעוד שמונה שעוקב אחר רצפים אחרים ייקרא בשמות אחרים (כגון מונה Gray וכו'). מונה בינארי בעל n רכיבי זיכרון יכול לספור מ-0 ועד ל- $2^n - 1$.

סוגי מונים:

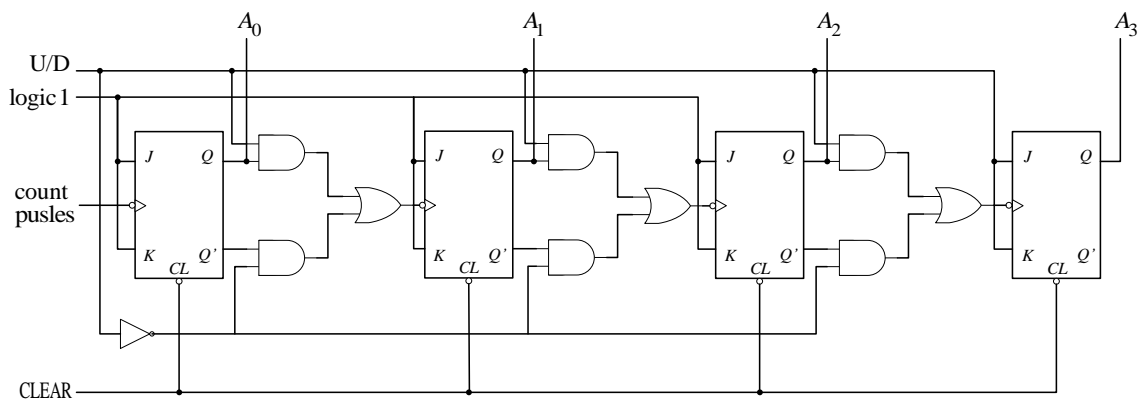
מונים נחלקים לשני סוגים מרכזיים:

- מונה אדווה (Ripple Counter) שהוא מונה **אסינכרוני** ביסודו. במונה מסוג זה, המוצא של רכיב זיכרון מתחבר לכניסת השעון של הדרגה הבאה, כך שכל דרגה מעוררת את הדרגה שאחריה.
- מונה סינכרוני (Synchronous Counter). במונה מסוג זה כל רכיבי הזיכרון מעוררים ע"י כניסת שעון משותפת. שני סוגים נפוצים של מונים סינכרוניים הם:
 - מונה טבעת (Ring Counter).
 - מונה טבעת מסובב (Johnson Counter).

מבנה כללי של מונה אדווה בינארי (Binary Ripple Counter):



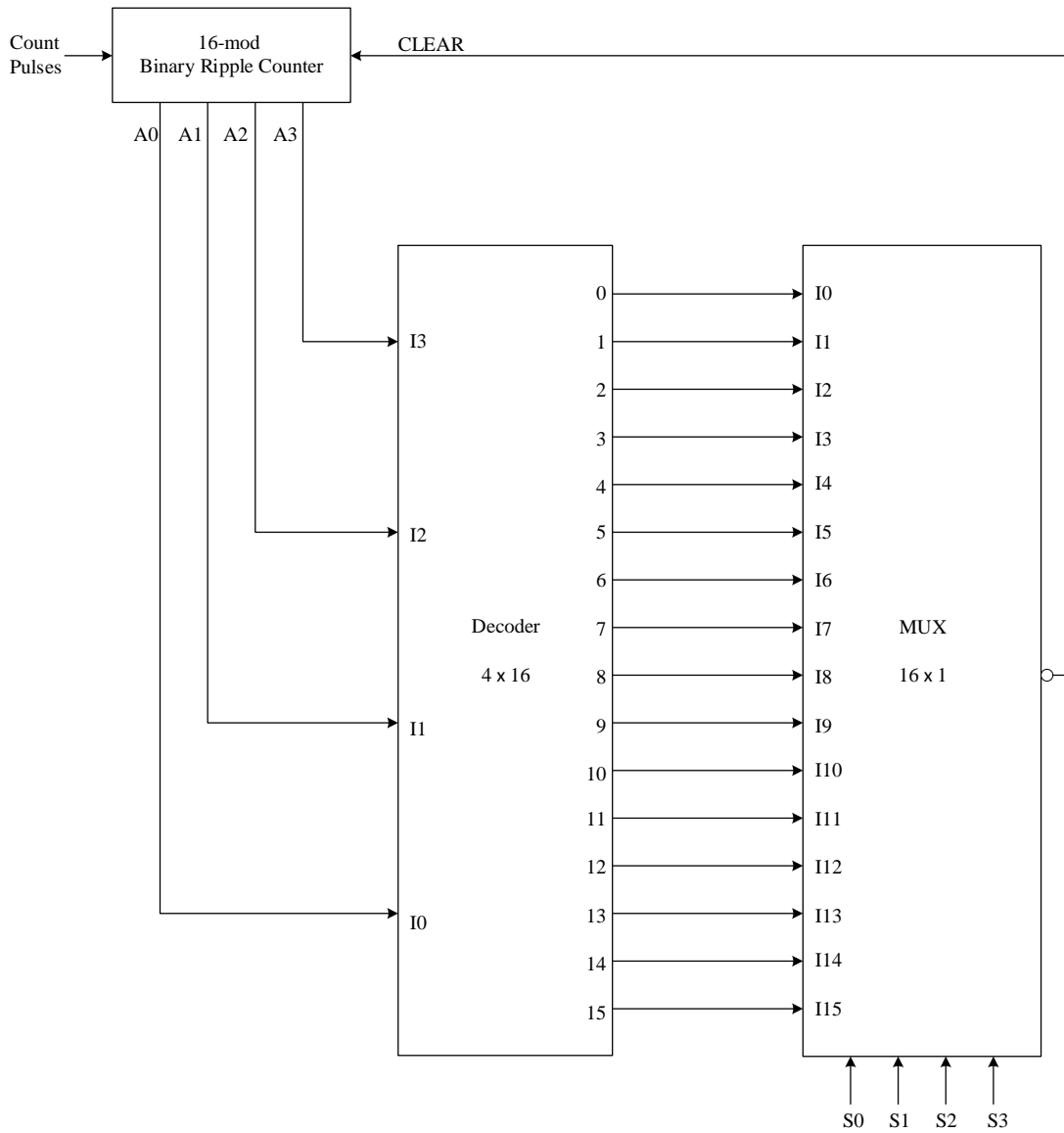
מבנה כללי של מונה הסופר מעלה ומטה (Up/Down Counter):



מונה עם ספירה לפי מודולו (modulo) מבוקש:

- נשתמש בכניסת CLEAR יחד עם שער AND/NAND מתאים לפי mod הספירה הרצוי.
- בהינתן $mod-M$, יש לבחור את מספרי רכיבי הזיכרון N לפי: $2^{N-1} < M \leq 2^N$.
- נוציא מ- $\log_2 N$ המוצאים את הערך המתאים ל- M על מנת שיאפס את כל רכיבי הזיכרון (תזכורת: $mod-M$ מאפשר ספירה של $[0 : M - 1]$).

הכללה של ספירה לפי כל mod מ-0 ועד ל- $M = 2^N - 1$ עבור $N = 4$:



אתחול מונים (Presettable Counters):

נשתמש בכניסת Preset של כל רכיבי הזיכרון על מנת לאתחל את המונה למספר נתון מראש לפי בקשה/דרישת המערכת.

מונים סינכרוניים:

מונים שבהם כניסת השעון מחוברת לכל רכיבי הזיכרון יחדיו.

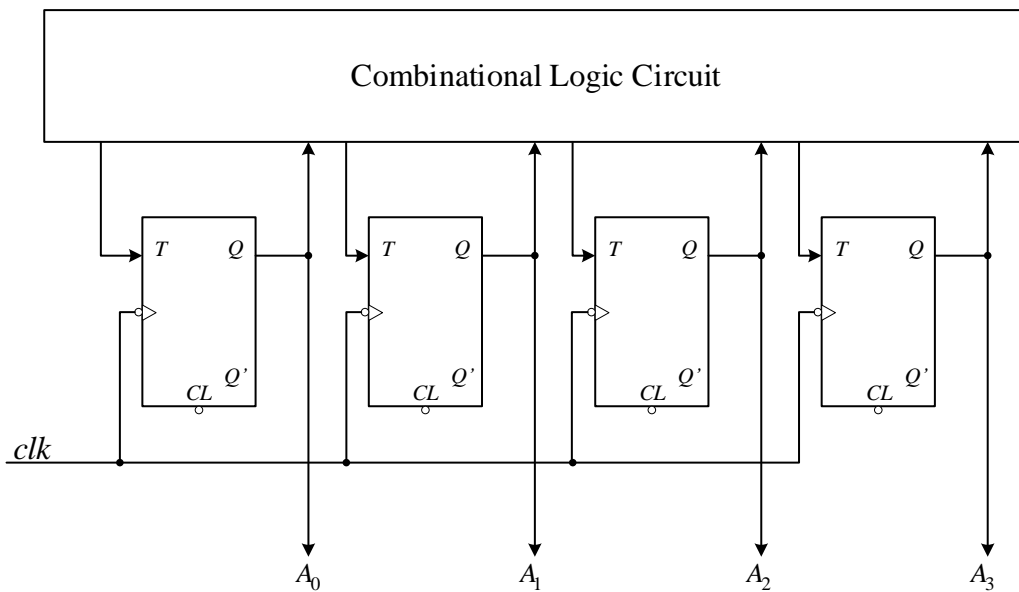
קיימים 3 סוגים מרכזיים של מונים סינכרוניים:

1. מונה סינכרוני בינארי/כללי.

2. מונה טבעת.

3. מונה ג'ונסון.

מבנה כללי של מונה סינכרוני:

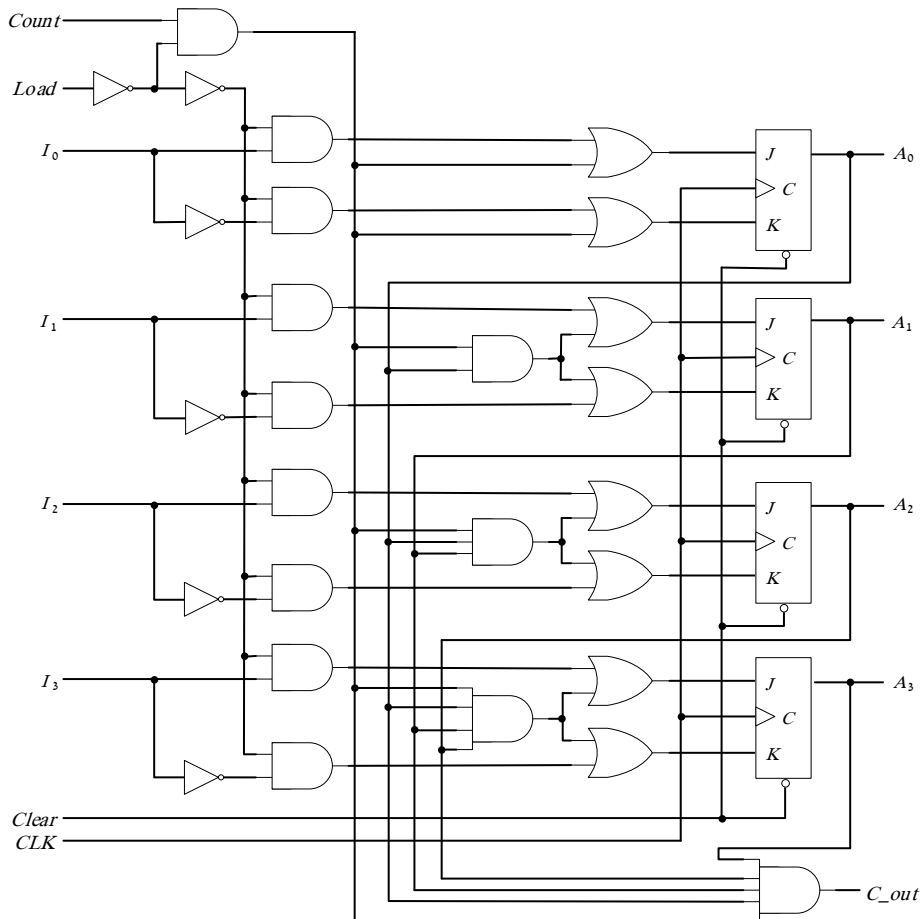


מונה (סינכרוני) בינארי - הגדרות ותכונות:

- חיבור של כל דרגה באמצעות שער AND לכל הדרגות שמתחתיה יניב ספירה בינארית של המונה.
- ע"י שימוש בלוגיקה ניתן להפוך את המונה לספירה מעלה ומטה.
(ראה בסרטון הוידאו מימושים והסברים).

אתחול מונה סינכרוני:

כדי לאתחל מונה סינכרוני נעזר בכניסת LOAD וע"י מעבר ל-1 לוגי נכניס את ערך המילה שבכניסת הרכיב $I_3I_2I_1I_0$. כמו כן נעזר בכניסת CLEAR כדי לאפס את המונה.

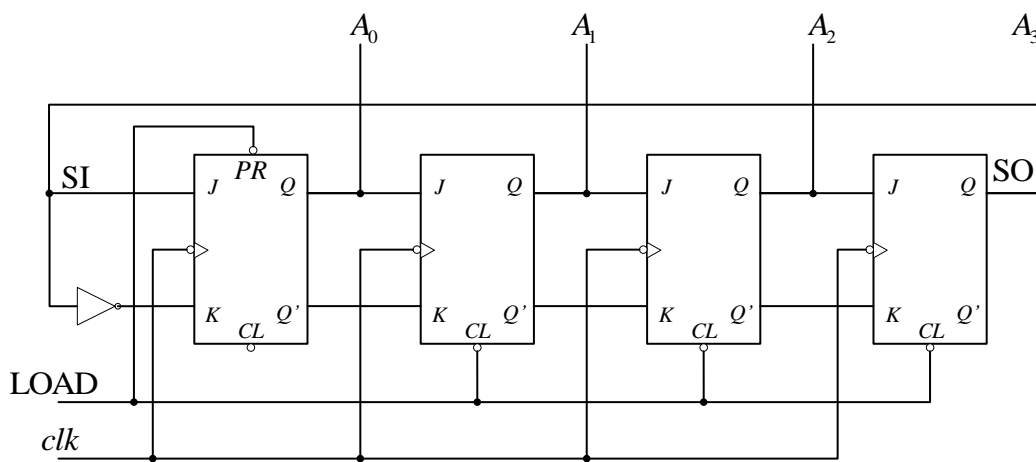


מונה עם מצבים ללא שימוש (Unused States):

נוכל לתכנן מונה שיעקוב אחר סדרת ערכים, לאו דווקא בינארית. במונה שכזה, יתכנו מצבים שלא יהיו בשימוש. יש לדאוג כי במידה והמונה נכנס למצב שאינו בשימוש הוא יחזור למצב תקין (self-starting).

מונה טבעת (Ring Counter):

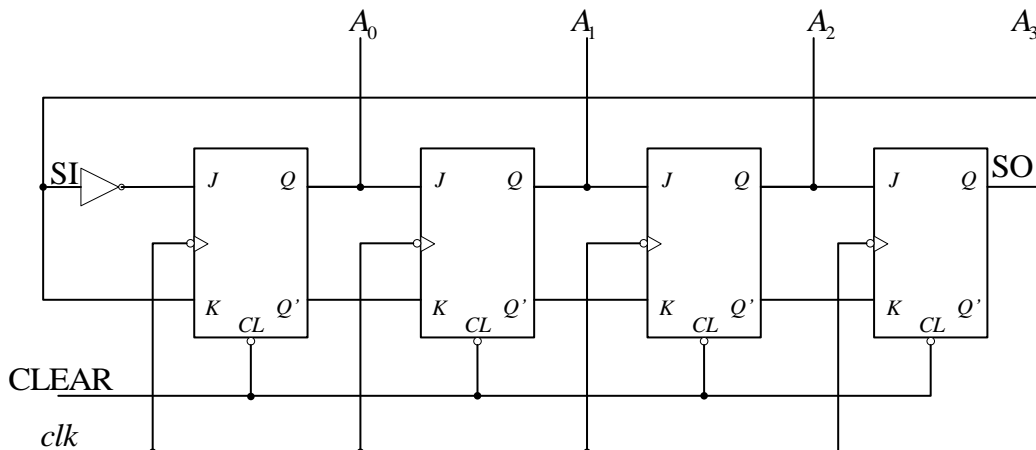
מונה המורכב כאוגר הזזה שבו המוצא מחובר לכניסה (SISO) באופן הבא:



מאתחלים את המונה בערך לוגי של 1 רק ברכיב זיכרון אחד (הנמוך ביותר) והוא 'עובר' מאחד לשני. יש לדאוג כי המונה לא יעבור למצב פעילות לא תקין אחרת הוא לא יחזור לפעילות תקינה.

מונה ג'ונסון (Johnson Counter):

נקרא גם מונה switch-tail או מונה twisted-ring ובנוי כמו מונה טבעת רק שכעת מחברים את המשלים של המוצא לכניסה:



שאלות:

(4) ענה על הסעיפים הבאים :

- א. ממש מונה אדווה (Ripple counter) בסיסי, הסופר מ-0 ועד ל-15 בעזרת 4 רכיבי זכרון מסוג T-FF. כלול כניסת Count pulses וכניסת Reset. הסבר את שיקולך.
- ב. ממש מונה אדווה (Ripple counter) בסיסי, הסופר מ-0 ועד ל-15 בעזרת 4 רכיבי זכרון מסוג D-FF. כלול כניסת Count pulses וכניסת Reset. הסבר את שיקולך.

(5) תכנן וממש מונה סינכרוני של קוד 2421 באמצעות רכיבי זיכרון מסוג T-FF.

(6) תכנן וממש מונה סינכרוני של קוד Excess-3 באמצעות רכיבי זיכרון מסוג D-FF.

- (7) יש לתכנן מונה מודולו 8 הסופר לפי הסדרה הבאה : $0, 5, 7, 6, 2, 3, 4, 1, 0, 5, 7, 6, 2, 3, 4, 1, \dots$ (הקריאה היא משמאל לימין, כלומר מ-0 עוברים ל-5, מ-5 ל-7 וכן הלאה).
 א. כתוב טבלת מצבים מתאימה (הגדר Q_2, Q_1, Q_0 כאשר Q_2 הוא MSB).
 ב. תן משוואות מתאימות עבור מימוש באמצעות יחידות של SR-FF.
 ג. ממש את המעגל (כלול כניסות CP, CL).

(8) ענה על הסעיפים הבאים :

- א. תכנן מונה בינארי מעלה מודולו 8 (Modulo 8 binary up counter) בעזרת יחידות זיכרון מסוג T-FF. הסתפק בכתיבת משוואות הכניסה לכל FF וציין האם יש צורך בשערים לוגים כלשהם, אם כן אילו?
- ב. לרשותך Mux 4×1 עם שני כניסות ברירה S_1, S_0 ויחידת 8 $\rightarrow 3$ Decoder. היעזר במונה שתכנתת בסעיף א' ובנה מעגל (הצג מימוש) הפועל לפי הטבלה הבאה :

פעולה	ערך כניסה S_1, S_0
מונה בינארי מעלה מודולו 2	00
מונה בינארי מעלה מודולו 5	01
מונה בינארי מעלה מודולו 6	10
מונה בינארי מעלה מודולו 7	11

- ג. היעזר במונה מסעיף א' כדי לתכנן ולממש מעגל הפועל לפי הטבלה הבאה. שים לב - לרשותך 3 שערי XOR ו-2 יחידות HA בלבד.

פעולה	ערך כניסה S_1, S_0
מונה של סעיף א' ללא שינוי	00
מונה בינארי מטה מודולו 8 לפי : 11, 10, ..., 4	01
מונה בינארי מעלה מודולו 8 לפי : 4, 5, 6, ... 11	10
מונה בינארי מטה מודולו 8	11

- (9) בשאלה זו נבנה מונה Ripple counter מודולו 7 עם JK-FF שעובד לפי העיקרון הבא:
- לרכיב כניסה אסינכרונית Up/Down שתסומן U/D.
 - כאשר אות $U/D = 1$ בתחילת הספירה, המונה מתאפס ואחרי זה סופר מעלה לפי סדר רגיל.
(שימו לב: כשהמונה מגיע לספרה 6 הוא עובר איפוס חזרה ל-0).
 - כאשר אות $U/D = 0$ בתחילת הספירה המונה נטען במספר 6 ואחרי זה סופר מטה.
(שימו לב: כאשר המונה מגיע לספרה 0 הוא עובר אתחול לספרה 6).
- תכנן את כניסת ה-Clear ואת כניסת ה-Preset בהתאם לדרישות המערכת.
צייר את המעגל הסופי.

תשובות סופיות:

- (1) עיין מימושים בסרטון הוידאו.
- (2) משוואות הכניסה: $T_0 = 1$; $T_1 = Q_0 + Q_2 \bar{Q}_3$; $T_2 = Q_0 Q_1 + Q_2 \bar{Q}_3$; $T_3 = Q_0 Q_1 Q_2 + Q_2 \bar{Q}_3$
- (3) משוואות הכניסה:
- (4) א. ראה טבלה בסרטון הוידאו.
ב. להלן המשוואות:
- $$D_0 = \bar{Q}_0; D_1 = \bar{Q}_0 \bar{Q}_1 + Q_0 Q_1; D_2 = \bar{Q}_0 \bar{Q}_1 Q_2 + Q_0 \bar{Q}_2 + Q_1 \bar{Q}_2; D_3 = Q_0 Q_2 + Q_1 Q_2 + Q_3$$
- ג. ראה מימוש בסרטון הוידאו.
- (5) א. $T_0 = 1$; $T_1 = Q_0$; $T_2 = \bar{Q}_0 Q_1$
ב. ראה בסרטון הוידאו.
ג. ראה בסרטון הוידאו.
- (6) ראה שיקולי תכנון ומימוש מלא לפי שלבים בסרטון הוידאו.