28/02/08

הרצאה 6- מבנה מחשבים

המרת אות אנלוגי לאות דיגיטלי

אות אנלוגי- אות רציף, כלומר שהערכים שהוא מקבל יכולים להיות כל מספר ממשי (כמו גובה, משקל, טמפרטורה, לחץ, כוח וכו')

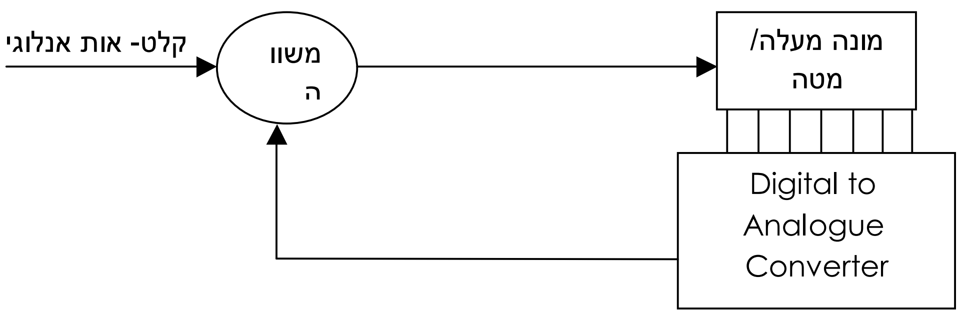
אות דיגיטלי- אות שהערכים שהוא יכול לקבל הם מספרים בדידים, כלומר לא כל מ"ס ממשי. יש לשים לב כי לא כל המספרים הממשיים ניתנים לייצוג במחשב, לכן המחשב "מעגל" את המספרים, ואז המספרים הממשיים במחשב הם בעלי ערכים בדידים (ע"ע מבוא לאנליזה נומרית).

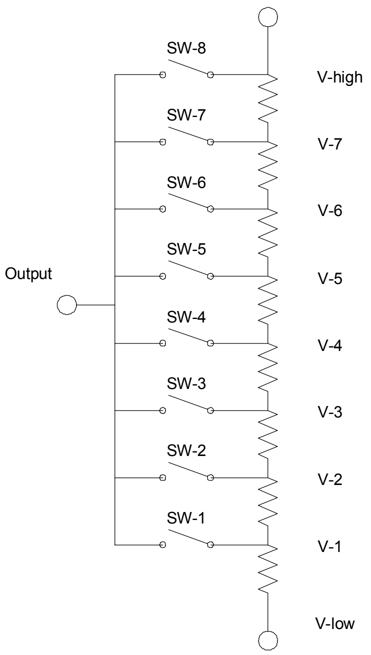
DSP (Digital Signal Processing) פירושו המרת אות אנלוגי לאות דיגיטלי, כלומר יצירת אות דיגיטלי שייצג אות אנלוגי, כדי שנוכל לבצע עליו כל מיני עיבודים או חישובים. למשל הקלטת קול למחשב- קול הוא אות אנלוגי, המחשב יודע לעבוד רק עם אותות דיגיטליים. לכן נרצה לבצע את ההמרה. לעיתים קרובות, נרצה גם לבצע את ההמרה ההפוכה (Digital to Analogue).

ההמרה מאנלוגי לדיגיטלי מתבצעת ע"י בחירת פרק זמן Δt. מחלקים את האות האנלוגי לקטעים שאורכם Δt באמצע כל קטע כזה אנו מבצעים דגימה של האות האנלוגי. באמצעות סדרת הדגימות הזו אנו יוצרים האות הדיגיטלי.

ההמרה מאנלוגי לדיגיטלי מתבצעת באמצעות רכיב בשם ADC (Analogue to Digital Converter) וההמרה ההפוכה מתבצעת באמצעות רכיב בשם DAC (Digital to Analogue Converter).

מימוש:





הסבר: פעולתו של הממיר עובדת בצורה שמזכירה קצת חיפוש בינארי במערך ממוין (או את המשחק חם-קר). ראשית אנו מתחילים עם מתח התחלתי שאותו אנו יוצרים בסולם נגדים (בציור). אפשר לייצג את הסולם הזה באמצעות מספר בינארי, כאשר כל מתג מייצג ביט (ספרה) מהמספר הבינארי, כך שכאשר מתג כלשהו מחובר, נאמר שהביט שהוא מייצג הוא 1, אחרת הביט הוא 0. הMSB מיוצג ע"י המתג העליון- sw-8 והLSB מיוצג ע"י SW-1 (המתג התחתון ביותר). לפי חוקי אוהם וחוקי קירכהוף, ככל שהמספר הבינארי המיוצג באופן זה יותר גדול גם כך גם המתח ביציאה מהסולם יותר גדול (אם אתם ממש מתעקשים להבין לכו לויקיפדיה).

את המתח אנו הופכים לאות אנלוגי באמצעות Digital-to-Analogue converter (אז מה קודם למה- התרנגול או הביצה?) ו (בלי להיכנס ליותר מדי פרטים) משווים אותו לאות האנלוגי בinput.

המשווה שלנו יודע להבדיל בין מצבים שבהם המתח של האות ההתחלתי היה גדול, קטן, או שווה למתח של האות בINPUT. אם הוא גדול אז הערך במונה נעלה/מטה עולה, אם הוא קטן אז הערך במונה קטן, ואם הם שווים אז הערך במונה לא משתנה.

המספר במונה, הוא בעצם המספר שמיוג ע"י סולם הנגדים.

שוב אנו ממירים את המספר שמיוצג ע"י סולם הנגדים לאות אנלוגי, משווים אותו אם האות INPUT וחוזר חלילה... עד שהמתחים פחות או יותר שווים. נתעלם מהפרטים.

אם לא הבנתם, תציצו בקישור

http://arstechnica.com/articles/paedia/audiofile-analog-to-digital-conversion.ars/2

ובטוח תבינו. יש שם גם דוגמה והוא כתוב בצורה מעניינת.

Finite State Machine

למרות שאת כל המעגלים האלקטרונים ניתן לייצג בצורה בוליאנית (קלט בוליאני, פלט בוליאני), לפעמים יותר קל לנו להבין את הדברים או לממש דברים אם אנו קודם מציירים אותם באמצעות מודל לא בוליאני. FSM (מכונת מצבים סופית) היא דוגמה למודל כזה.

כל מעגל אלקטרוני שיש בו רכיבים עם זיכרון הוא FSM, כאשר מצב הוא בעצם הערכים המאוחסנים בזיכרון.

מכונות מצבים קטנות נבנות ע"פ השלבים הבאים:

- הגדרת מצבים
- הגדרת פונקצית מעברים (כלומר פונקציה שבהינתן המצב הנוכחי והקלט מחליטה מהו המצב הבא)
- צמצום מצבים (מינימיזציה).

בדיוק כמו בקורס "מודלים חישוביים" (למי שעדיין לא שם לב...).

עבור כל מצב יש מספר סופי של מצבים עוקבים. המצב הנוכחי הוא המצב אליו הגענו במחזור השעון הנוכחי, ולפי הקלט שמתקבל במחזור השעון אנו מחליטים מהו המצב שנהיה בו במחזור השעון הבא (לפי פונקציית המעברים של המכונה). יכול להיות שעבור מצב מסוים, מתקיים שאחד המצבים העוקבים שלו הוא המצב עצמו (שוב, לפי מה שמוגדר בפונקציים המעברים).

אנו דנים בשני סוגים של מכונות מצבים:

Mealy machine:

הפלט תלוי במצב הנוכחי בלבד או במצב הנוכחי ובקלט הנוכחי. (לכן כתוב על החץ שמתאר את המעבר למצב הבא).

:Moore Machine

הפלט תלוי במצב הנוכחי בלבד. (לכן כתוב בתוך העיגול שמתאר את המצב).

הבדלים:

- תמיד נוכל לתאר את אותה הפונקציה הן באמצעות מכונת Mealy ומכונת Moore (הן שקולות), אבל בד"כ תיאור באמצעות מכונת Mealy כרוך בפחות מצבים, לכן המימוש יתפוס פחות מקום.
- מכונת Mealy מגיבה (מוציאה פלט) ישירות כשהיא מקבלת את הקלט, בניגוד למכונת Moore שמוציאה פלט רק במצב הבא. כלומר מכונת Mealy מגיבה מחזור שעון אחד לפני מכונת Moore.
- מכונת Moore היא יותר יציבה, היא אינה משנה את הפלט במשך מחזור השעון (כי הפלט תלוי במצב בלבד), לעומת זאת Mealy עלולה לשנות את הפלט שלה במשך מחזור השעון, אם הקלט לא יציב (כי בmealy הפלט גם תלוי בקלט). תציצו בשקפים 18-16 בהרצאה על FSM.

דוגמה (מההרצאה)- מכונת מצבים שהפלט שלו הוא 1 ברגע שהתקבלו 3 1-ים ברצף, ו-0 אחרת.

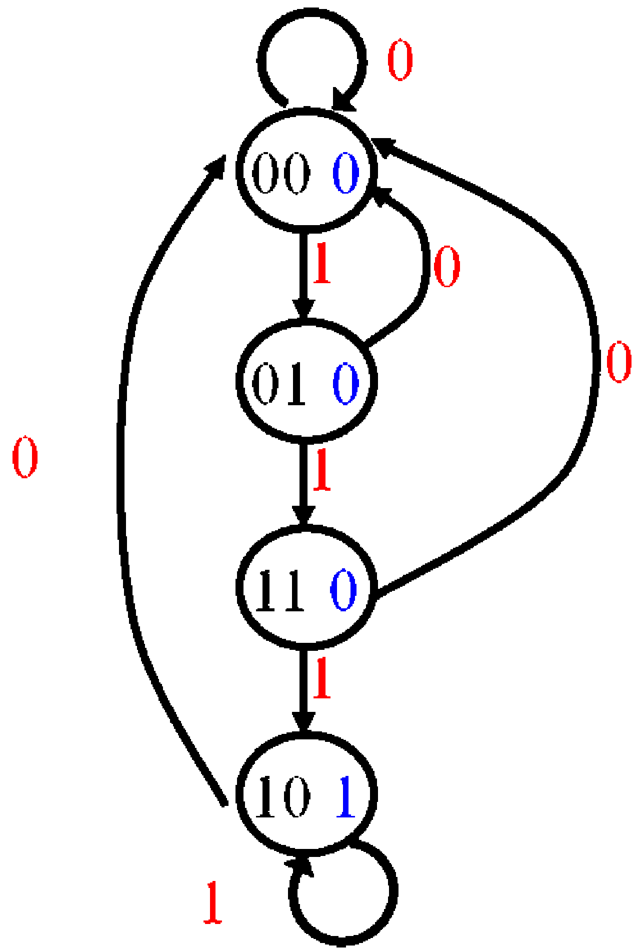
מימוש באמצעות מכונת Moore:

התרשים:

בשחור- מספר המצב

באדום- הINPUT

בכחול-הOUTPUT



טבלת המצבים:

I הוא הINPUT,

A_tB_t הם שני הביטים שמייצגים את מ"ס המצב בזמן t, יש 4 מצבים כפי שרואים בשחור בתרשים לכן הם מיוצגים באמצעות שני ביטים,

O_t זה הOUTPUT בזמן t.

		I=0		I=1		
A_t	B_t	A_{t+1}	B_{t+1}	A_{t+1}	B_{t+1}	O_t
0	0	0	0	0	1	0
0	1	0	0	1	1	0

1	1	0	0	1	0	0
1	0	0	0	1	0	1

מימוש:

מכיוון שיש שני ביטים שמייצגים את מספר המצב, נהיה זקוקים לשני FF. נממש באמצעות שני FF מסוג DATA (היינו D-FF). נסמנם A וB.

A_t-ייצג את הכניסה D של A.

B_t - ייצג את הכניסה D של B.

A_{t+1} - ייצג את היציאה Q של A.

B_{t+1} - ייצג את היציאה Q של B.

נעשה מפות קרנו הן עבור A_{t+1} ועבור B_{t+1} (לכל אחד בנפרד), נזכור שהמצב $A_{t+1}B_{t+1}$ תלוי הן ב INPUT והן במצב הקודם A_tB_t, לכן מפות הקרנו יהיו בנויות באופן הבא:

(לשם הפשטות נרשום מעכשיו A במקום A_t וB במקום B_t).

AB

	A_{t+1}	00	01	11	10
I	0				
	1				

עבור A_{t+1}, וכן

AB

	A_{t+1}	00	01	11	10
I	0				
	1				

עבור B_{t+1}.

אחרי הפישוטים שנעשה באמצעות מפת קרנו נקבל כי

$$A_{t+1} = A*I + B * I = I(A+B)$$

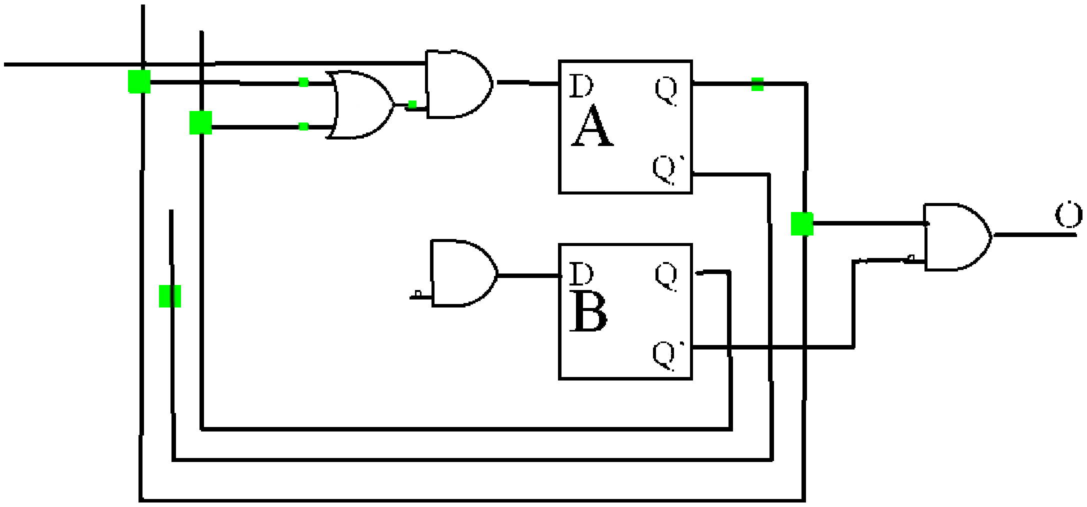
וכן

$$B_{t+1} = A*I$$

וקל לראות שמתקיים

$$O = A*\overline{B}$$

נצייר את המעגל:



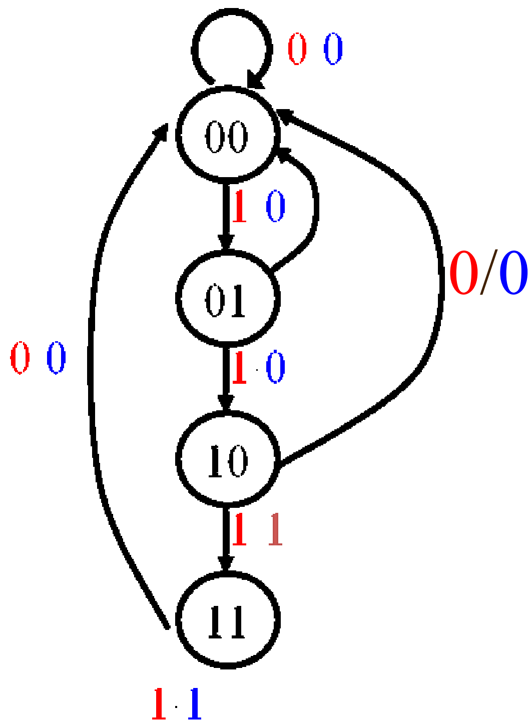
כעת נחזור חלילה עבור מכונת Mealy:

התרשים:

בשחור- מספר המצב

באדום- הINPUT

בכחול-הOUTPUT



טבלת המצבים:

I הוא הINPUT,

A_tB_t הם שני הביטים שמייצגים את מ"ס המצב בזמן t, יש 4 מצבים כפי שרואים בשחור בתרשים לכן הם מיוצגים באמצעות שני ביטים,

O_t זה הOUTPUT בזמן t.

		X=0		X=1		X=0	X=1
A_t	B_t	A_{t+1}	B_{t+1}	A_{t+1}	B_{t+1}	O_t	O_t
0	0	0	0	0	1	0	0
0	1	0	0	1	0	0	0
1	0	0	0	1	1	0	0
1	1	0	0	1	1	0	1

מימוש:

מכיון שיש שני ביטים שמייצגים את מספר המצב, נהיה זקוקים לשני FF. נממש באמצעות שני FF מסוג DATA (היינו D-FF). נסמנם A וB.

A_t-ייצג את הכניסה D של A.

B_t- ייצג את הכניסה D של B.

A_{t+1}- ייצג את היציאה Q של A.

B_{t+1}- ייצג את היציאה Q של B.

נעשה מפות קרנו הן עבור A_{t+1} ועבור B_{t+1} (לכל אחד בנפרד), נזכור שהמצב $A_{t+1}B_{t+1}$ תלוי הן ב INPUT והן במצב הקודם A_tB_t, לכן מפות הקרנו יהיו בנויות באופן הבא:

(לשם הפשטות נרשום מעכשיו A במקום A_t וB במקום B_t).

$\overbrace{\qquad\qquad\qquad\qquad}^{AB}$

	At+1	00	01	11	10
I	0				
	1				

עבור At+1, וכן

AB

	At+1	00	01	11	10
I	0				
	1				

עבור Bt+1.

אחרי הפישוטים שנעשה באמצעות מפת קרנו נקבל כי

$$A_{t+1} = X \bullet (A + \overline{B})$$

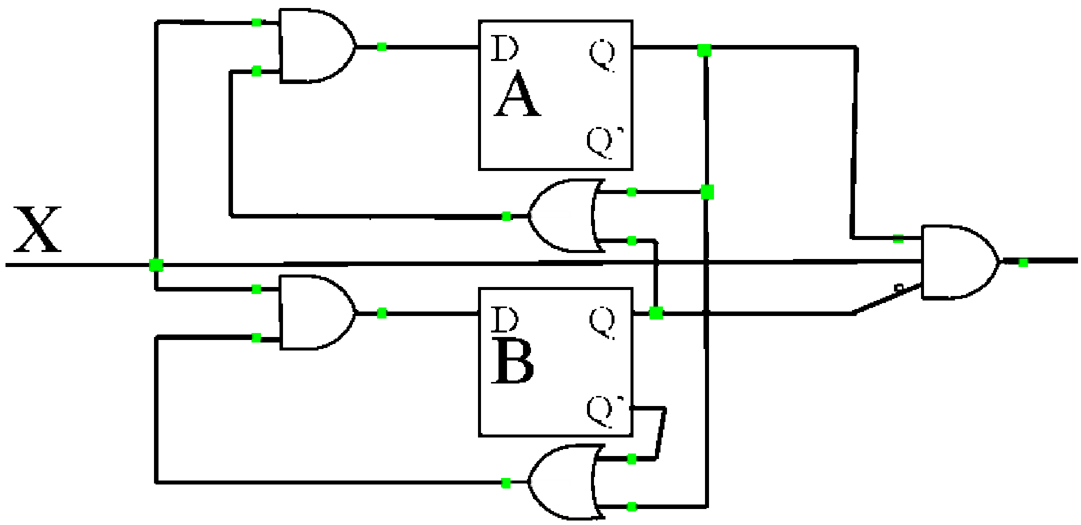
וכן

$$B_{t+1} = X \bullet (A + \overline{B})$$

וקל לראות שמתקיים

$$O = A \bullet B \bullet X \ .$$

נצייר את המעגל:


X
D
Q
A
Q'
D
Q
B
Q'