



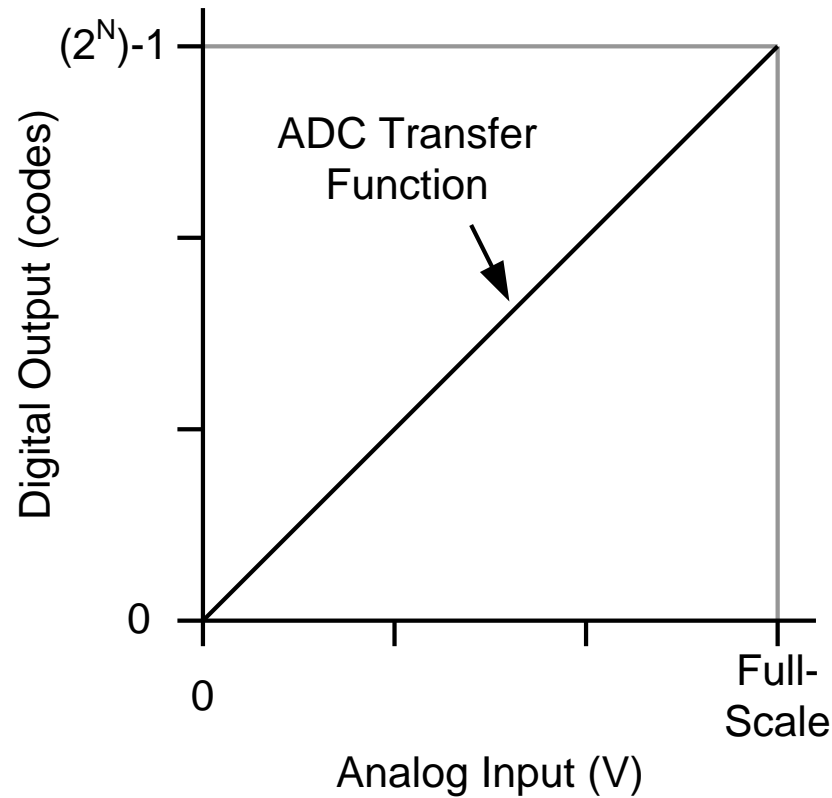
Lecture 12

Analog to Digital Converters

ממיר אנלוגי לספרתי. ADC.

- ◆ מה זה ADC?
- ◆ פלט מול קלט.
- ◆ טווח ערכים.
- ◆ צורות קלט: חד קוטבי מול דיפרנציאלי.
- ◆ צורת פלט חד קוטבי מול דו קוטבי.
- ◆ רכיבים אנלוגיים במיקרו בקר C8051F020
- ◆ ממיר ADC 12 סיביות (ADC0).
 - הפעלת המרה בעזרת ADC0
 - מילת נתונים של מפת המרה (12-bit)
 - תכנון ADC0
 - זיהוי של סיום תהליך המרה של ADC0
 - תדר המרה SAR0
 - דוגמא של שיטת polling בעזרת ADC0.
 - דוגמא של שיטת פסיקות בעזרת ADC0.
- ◆ 8-bit ADC (ADC1)

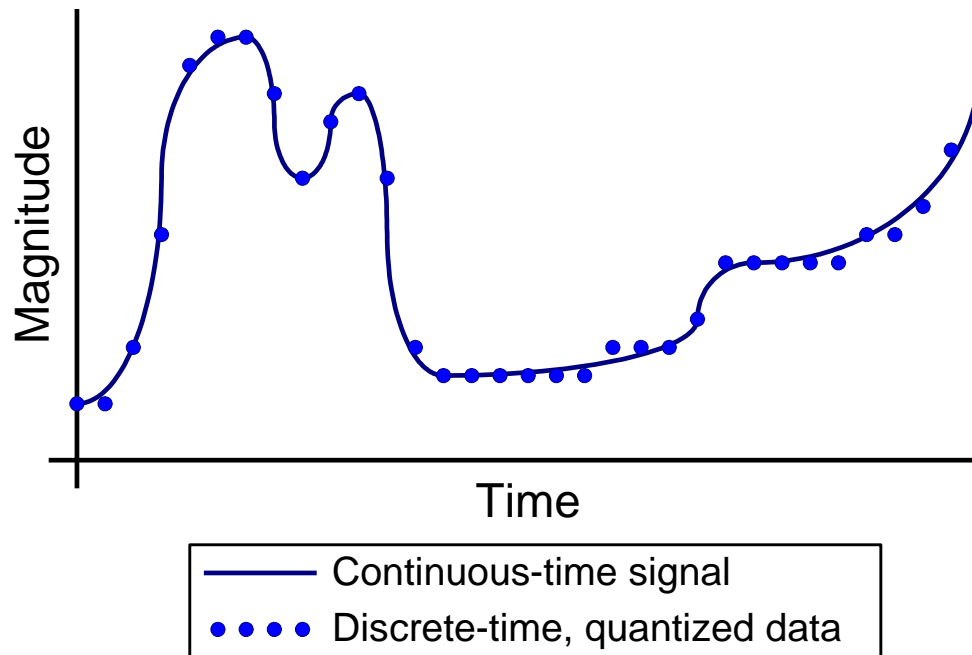




- ◆ ADC זה קיצור של "ממיר אנלוגי לדיגיטלי".
- ◆ ADC לוקח מתח אנלוגי בכניסה ומתרגם אותו למספר שמתאר מתח זה ביציאה.

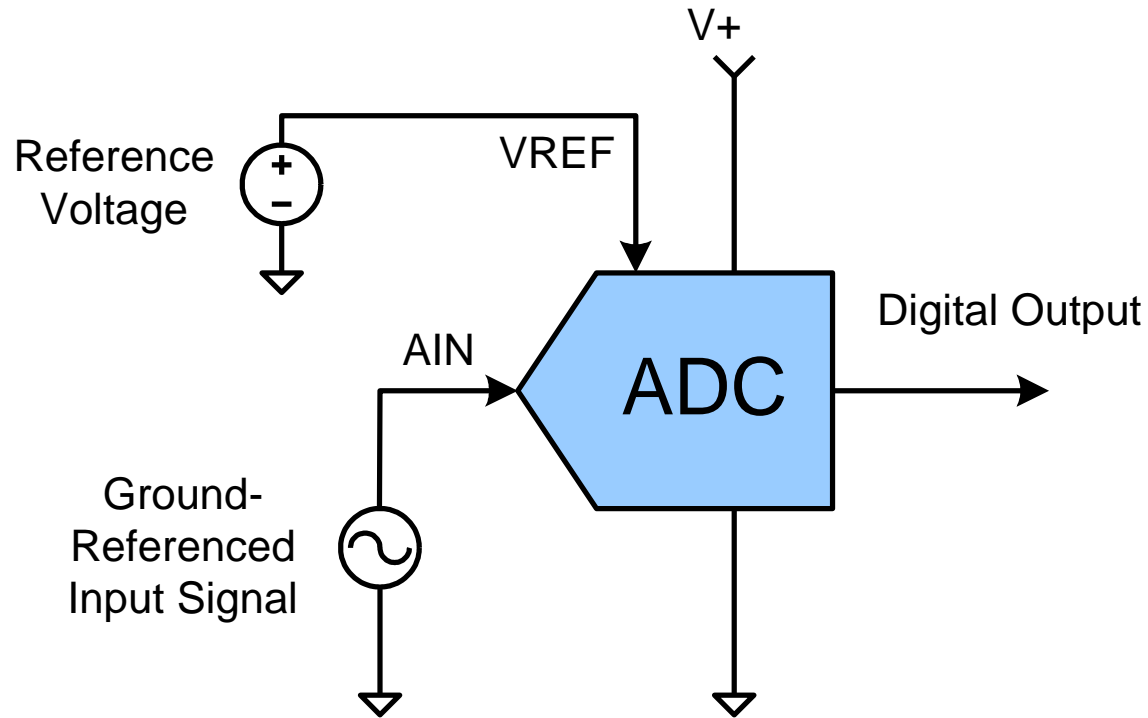
יש הבדל בין יציאה לכניסה: ♦

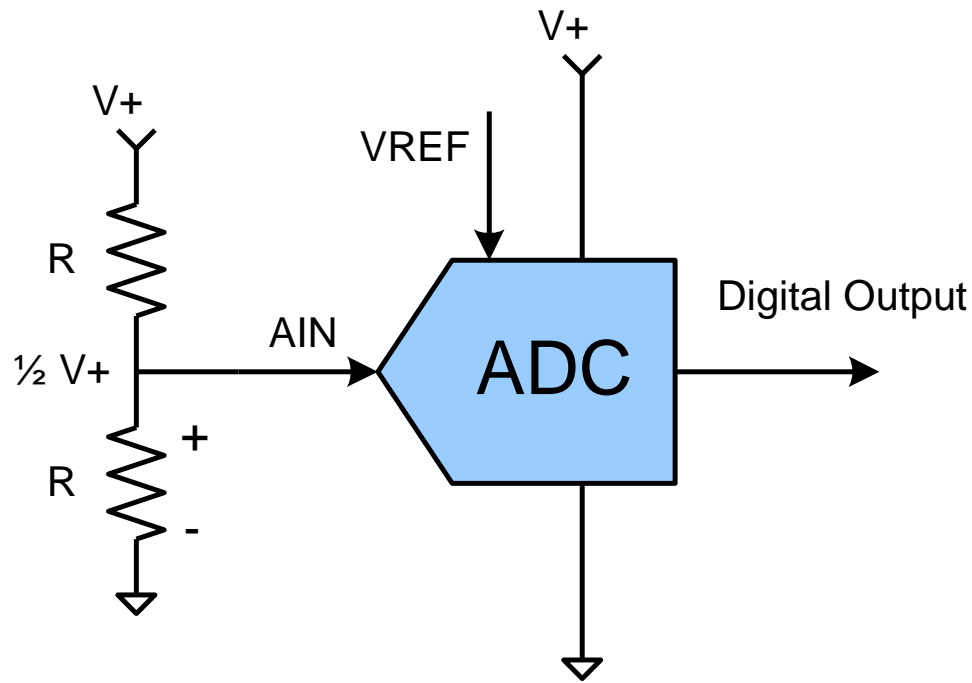
מתח אנלוגי בכניסה הוא מתח ממושך ומתח ספרתי ביציאה הוא מתח בדיד שמיוצג ע"י קודים ספרתיים.



- ◆ טווח ערכים בכניסה מוגדר ע"י מתח ייחוס (V_{REF}) שמסופק לממיר.
- ◆ מקור מתח גם חשוב לחישוב מתח מוחלט בכניסה.
 - ברוב ארכיטקטורות מתח כניסה מעבר לרמה של מתח יציאה לא ניתן למדידה ויכול לגרום לשריפה של הרכיב.

- ◆ ממיר חד קוטבי הוא ממיר שמתח כניסה שלו נמדד בין V_{ref} לבין אדמה (ביחס לאדמה).

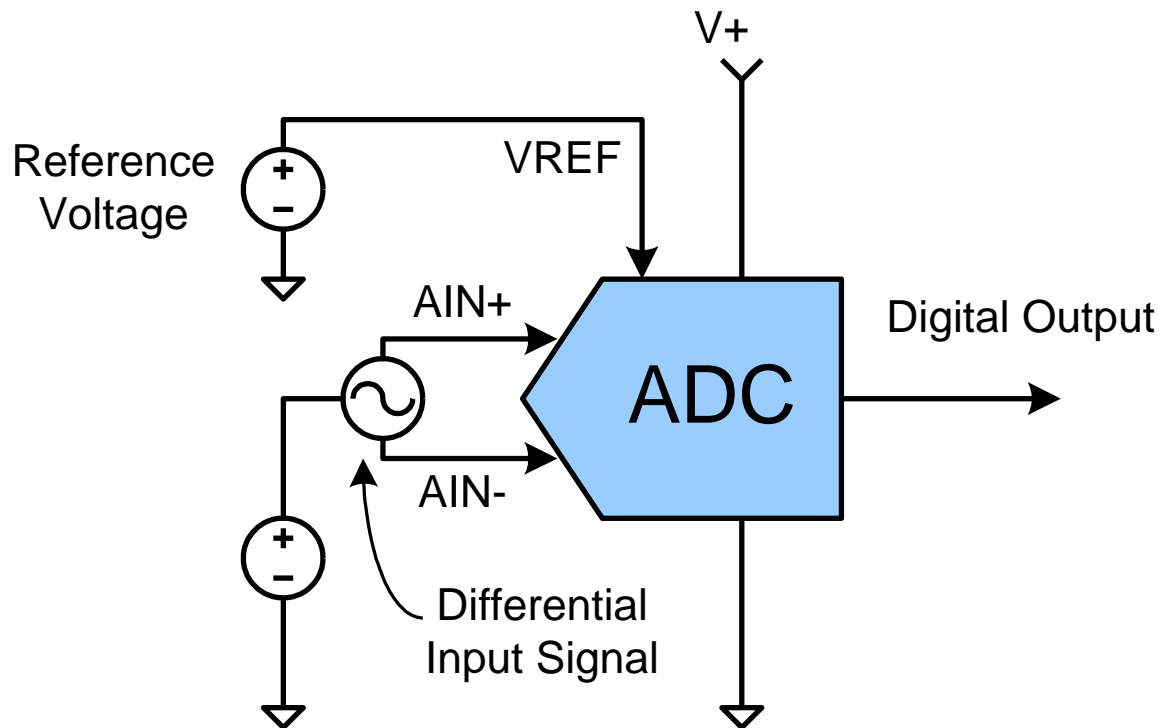




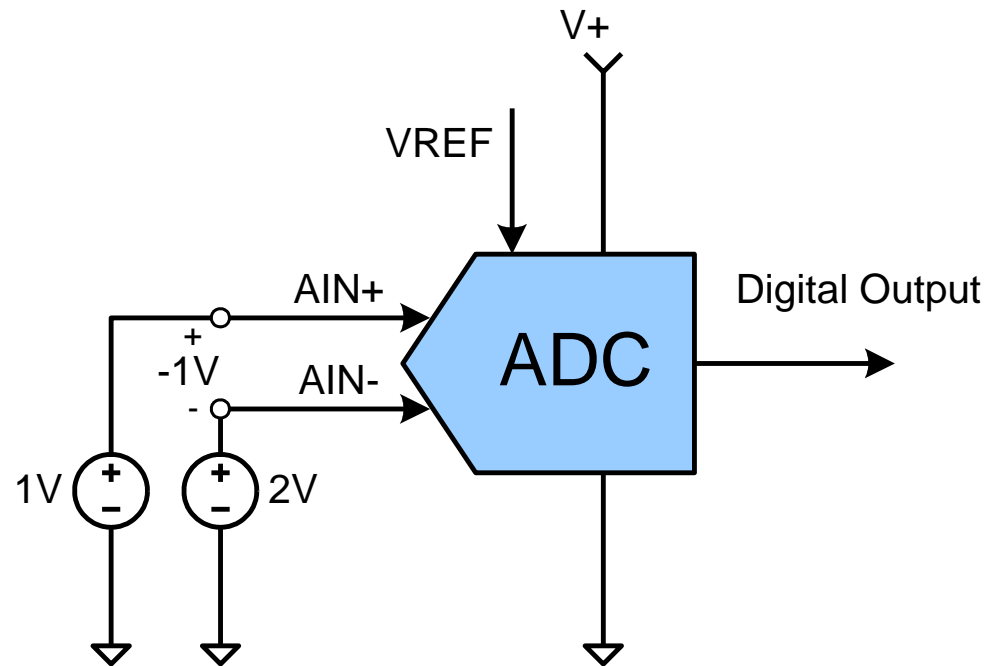
◆ דוגמא לחיבור מעגל של ADC חד קוטבי בעזרת מחלק מתח.

ADC דיפרנציאלי (הפרשי, דו-קוטבי).

- ◆ לממיר הפרשי נמדד הפרש מתחים בין שני פינים ($A_{IN+} - A_{IN-}$).
- טווח מתחי כניסה היא בין $-V_{REF}$ עד $+V_{REF}$ או פעמיים טווח של ממיר חד קוטבי.

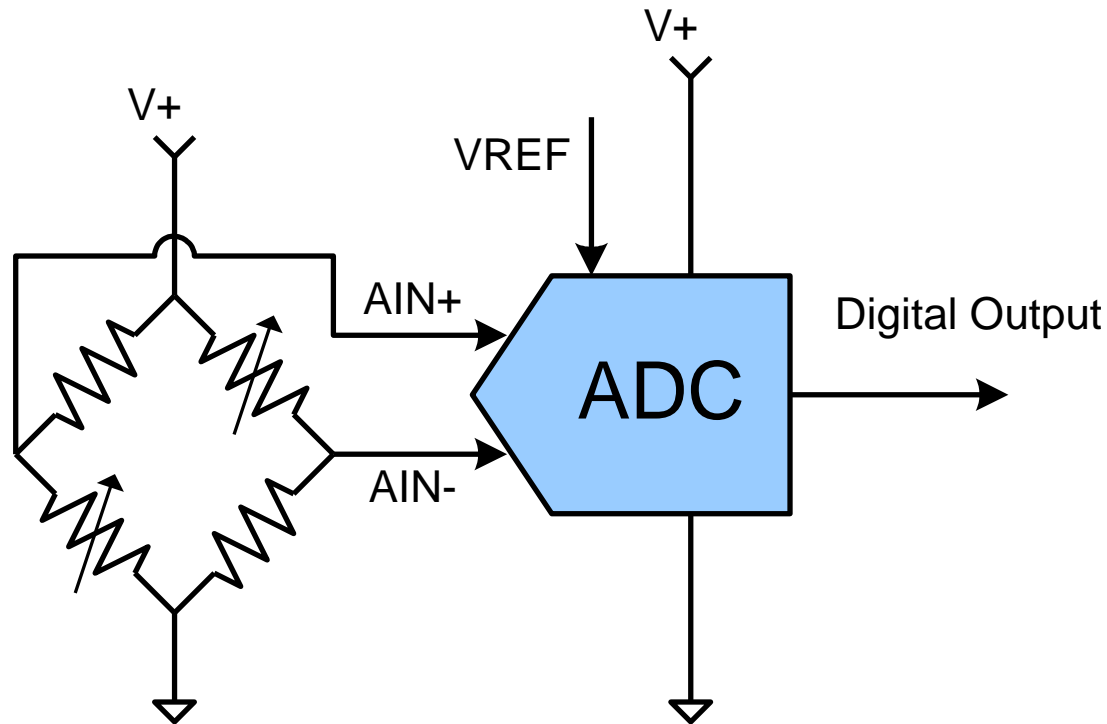


ADC דיפרנציאלי (הפרשי, דו-קוטבי).



- ◆ לא חייב להיות מתח שלילי בכניסה שלילית של ממיר. ADC
- ◆ אם הפרש בין מתחים A_{IN+} ו- A_{IN-} שלילי, ביציאה נראה מתח שלילי.
 - לדוגמא: $A_{IN+} = 1\text{ V}$, $A_{IN-} = 2\text{ V}$.
 - $(A_{IN+} - A_{IN-}) = (1\text{ V} - 2\text{ V}) = -1\text{ V}$

ADC דיפרנציאלי (הפרשי, דו-קוטבי). מדידה ע"י גשר נגדים.



◆ דוגמא של מדידת מתח הפרשי שנקבע ע"י גשר נגדים.

- ◆ טווח ערכים ביציאה של ממיר נקבע ע"י נוסחה 2^N , כאשר N – מספר סיביות של ממיר במילת פלט.
- ◆ כדי לחשב מצב ספרתי ביציאה צריכים לקחת בחשבון מתח כניסה, מתח ייחוס וכמות סיביות של המרה. ניתן לחשב מצב ביציאה ע"י נוסחה הבאה: (המתח בכניסה הוא חד קוטבי).
$$\text{Output} = (V_{IN} / V_{REF}) \times 2^N$$
- ◆ כדי לחשב מתח כניסה אנלוגי שדרוש בשביל מצב מסוים ביציאה, עושים שינוי נושא נוסחה:
$$V_{IN} = V_{REF} \times (\text{Output} / 2^N)$$
- ◆ משתמשים במושג "LSB", כדי לציין מהו ערך הקטן ביותר (ערך של סיבית בודדת) שניתן לקבל מממיר. ניתן לחשב אותו לפי נוסחה הבאה:
One LSB = input voltage range/output code range ➤
לדוגמא: ניקח ממיר חד קוטבי 12 סיביות, מתח ייחוס 2.4 V :
one LSB = $(V_{REF} / 2^{12}) = (2.4 \text{ V} / 4096) = 0.59 \text{ mV}$

קידוד ביציאה של ממיר חד קוטבי.

- ◆ משתמשים בקידוד חד קוטבי, כאשר אות כניסה ל-ADC חיובי.
- ◆ כאשר ממיר מסוג חד קוטבי, קידוד ביציאה גם חד קוטבי.
- ◆ כאשר קידוד ביציאה חד קוטבי, ממיר משתמש במספרים בינריים לא מסומנים.

Input Voltage	Output Code (12-bit)
$\geq V_{REF}$	4095 (0x0FFF)*
$V_{REF} - 1 \text{ LSB}$	4095 (0x0FFF)
$\frac{1}{2} V_{REF}$	2048 (0x0800)
$\frac{1}{4} V_{REF}$	1024 (0x0400)
0 V	0 (0x0000)
* ממיר נמצא במצב רוויה.	



קידוד ביציאה של ממיר דו קוטבי.

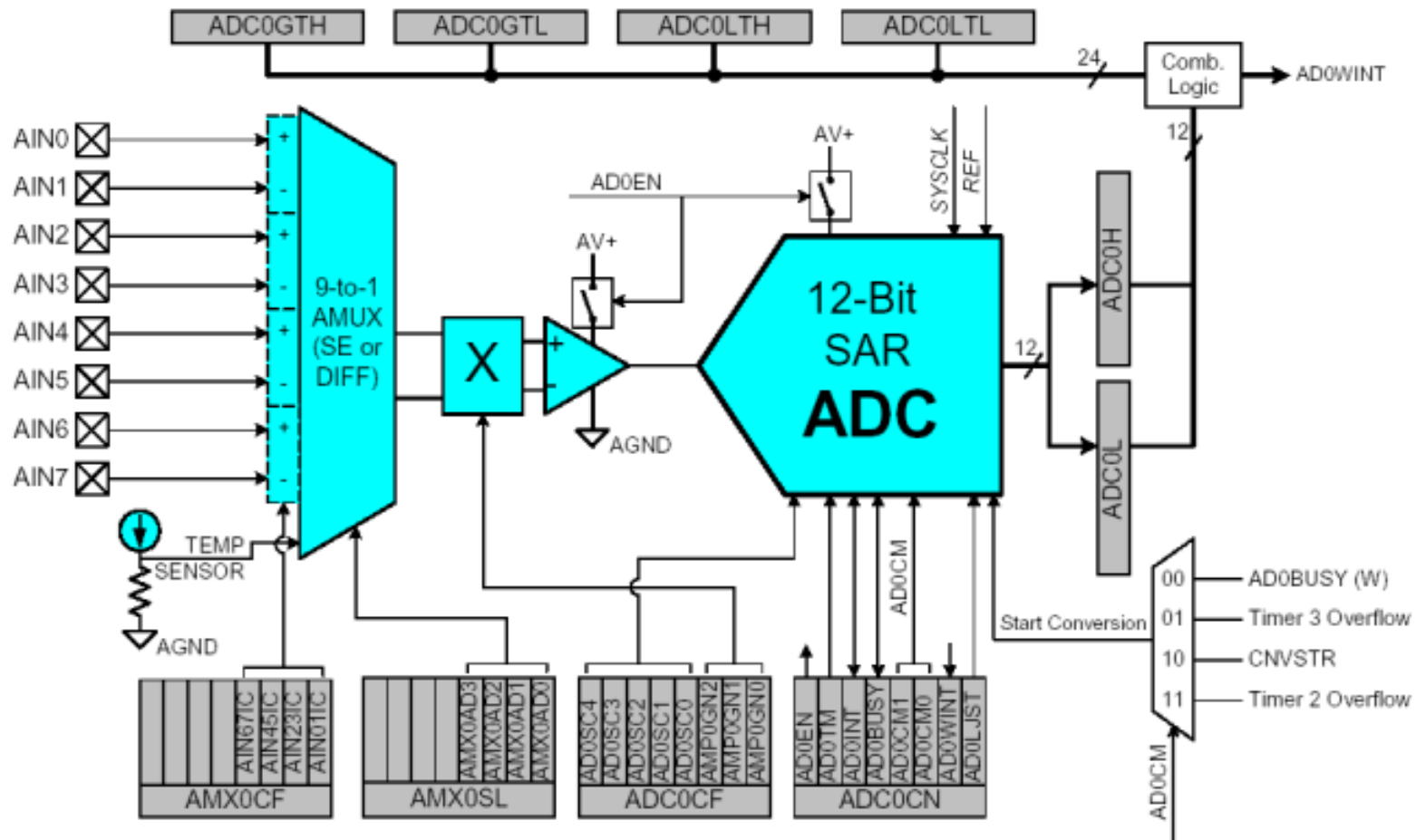
- ◆ משתמשים בקידוד דו קוטבי, כאשר הפרש מתחים בכניסה לממיר יכול להיות חיובי או שלילי. (ממיר הפרשי).
- ◆ לממיר הפרשי טווח בכניסה מוכפל וזה גורם להכפלת ערך של LSB.
- ◆ בד"כ משתמשים בקידוד "משלים ל-2" כדי להציג ערך דו קוטבי ביציאה.

Input Voltage	Output Code (12-bit, sign extended)
$\geq V_{REF}$	2047 (0x07FF)*
$V_{REF} - 1 \text{ LSB}$	2047 (0x07FF)
$\frac{1}{2} V_{REF}$	1024 (0x0400)
0 V	0 (0x0000)
$-\frac{1}{2} V_{REF}$	-1024 (0xFC00)
$-V_{REF}$	-2048 (0xF800)
$< -V_{REF}$	-2048 (0xF800)*
* ממיר נמצא במצב רוויה.	



- ◆ למיקרו בקר C8051F020 יש מספר רכיבים אנלוגיים:
 - ממיר ADC 12 סיביות וממיר ADC 8 סיביות.
 - 2 ממירים DAC 12 סיביות.
 - מגברים עם הגבר ניתן לשינוי ותכנון. (PGA)
 - מרבב אנלוגי (8 ו-9 ערוצים).
 - 2 משוויים אנלוגיים.
 - מתח ייחוס מדויק.
 - חיישן טמפרטורה.

12-Bit ADC (ADC0)



12-Bit ADC (ADC0)

◆ מערכת של ממיר ADC0 מכילה רכיבים הבאים:

➤ מרבב אנלוגי 9 ערוצים, ניתן לשינוי הגדרות. (AMUX0)

▪ 8 ערוצים לכניסות חיצוניות.

• כניסות חד קוטביות.

• זוגות של כניסות דו קוטביות.

▪ ערוץ 9 מיועד לחיישן טמפרטורה פנימי.

➤ מגבר עם הגבר ניתן לשינוי ותכנון. (PGA0)

▪ הגבר ברירת מחדל - 1

▪ ניתן להגביר אות פי 0.5, 1, 2, 4, 8 או 16.

➤ אוגר SAR 12 סיביות. Successive approximation register

◆ כדי לאפשר עבודת הממיר צריכים לעלות סיבית **AD0EN** (ADC0CN.7) ל-"1".



הפעלת המרה של ממיר ADC0.

◆ ניתן להתחיל המרה באחד מארבע דרכים שונות. תלוי במצבים של סיביות **AD0CM0** ו- **AD0CM1** של אוגר **ADC0CN**.

1. פקודה ברמת תוכנה ("1" ב-**AD0BUSY**).

2. מילוי יתר של טיימר 2.

3. מילוי יתר של טיימר 3.

4. אות חיצוני (פולס עלייה ב- **CNVSTR**).

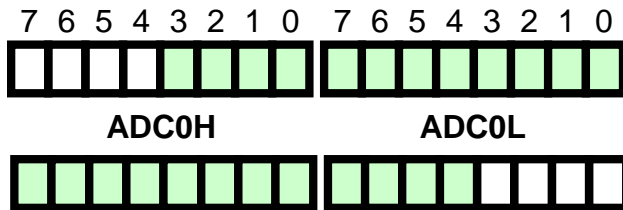
◆ סיבית **AD0BUSY** נשאר במצב "1" במשך זמן ההמרה והופך ל-"0", כאשר המרה מסתיימת.

◆ פולס ירידה של **AD0BUSY** מפעיל פסיקה (אם היא מאופשרת) ומעלה ל-"1" דגל פסיקה **AD0INT** (**ADC0CN.5**).

◆ במידה ופסיקה של "סיום ההמרה" מאופשרת (**EIE2.1**), אז תהיה פסיקה, כאשר **AD0INT** יהיה במצב "1" ואז תת-שגרה **ADC0 ISR** תתחיל להתבצע.



- ◆ מילה לאחר המרה נמצאת אוגרים **ADC0H** ו- **ADC0L**. מילה יכולה להיות צמודה לצד שמאל או לצד ימין, בהתאם לבקשה של משתמש. ישנה סיבית **AD0LJST** (**ADC0CN.0**) שהיא אחראית על צורת המילה:



➤ $ADC0H[3:0]:ADC0L[7:0]$, if $AD0LJST = 0$
($ADC0H[7:4]$ will be 0000b)

➤ $ADC0H[7:0]:ADC0L[7:4]$, if $AD0LJST = 1$
($ADC0L[3:0] = 0000b$)

- ◆ ניתן להיעזר בנוסחה הבאה כדי לחשב קוד ספרתי:

$$ADC0Code = Vin \times \frac{Gain}{VREF} \times 2^n$$

n=12 במידה וכניסה חד קוטבית או n=11 במידה וכניסה היא הפרשית (דו קוטבית).

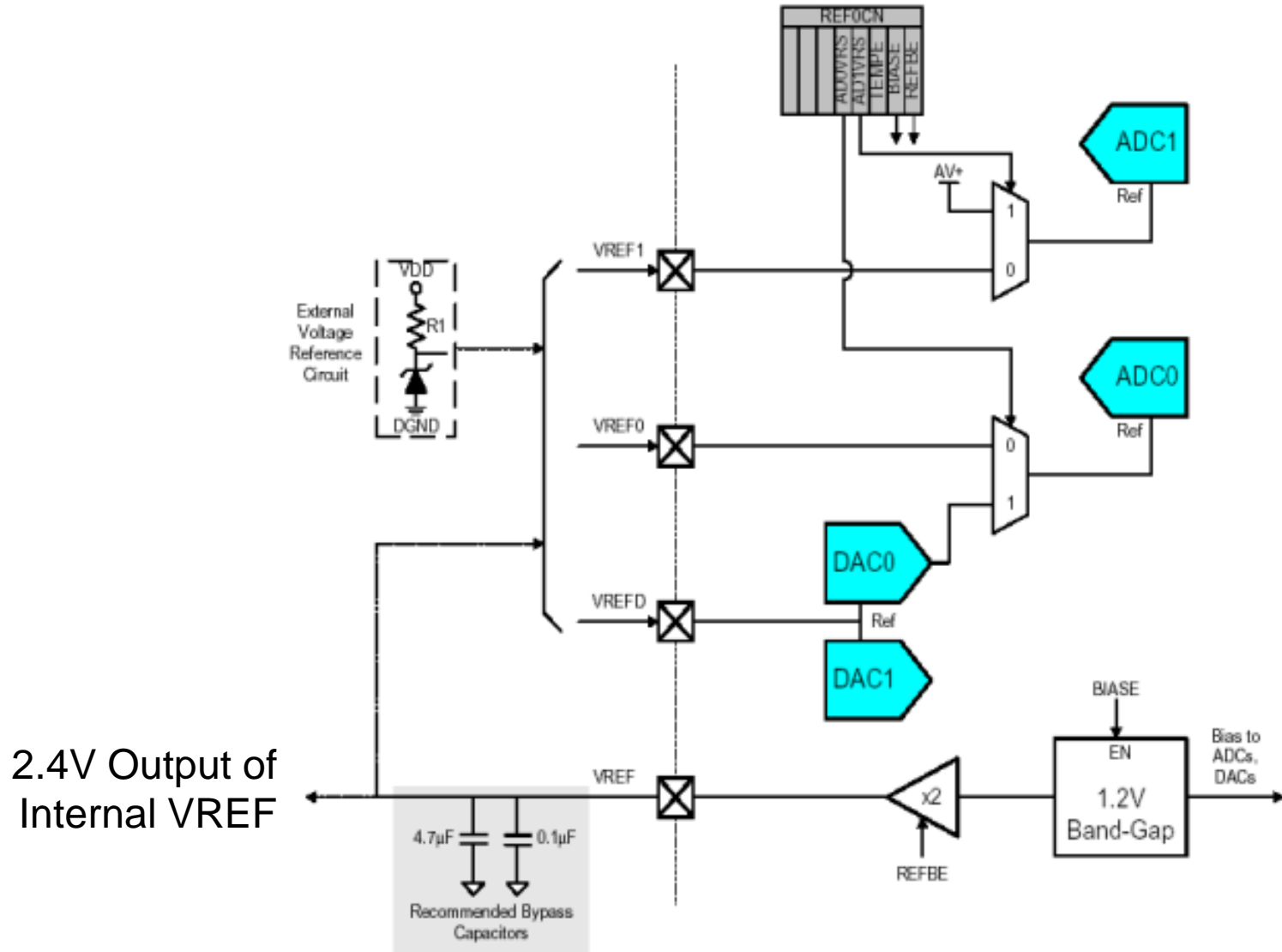
◆ נניח שAIN0 נמצא במצב חד קוטבי (AMXSL=00H ו- AMX0CF=00H) והגבר מוגדר ל- 1.

AIN0 – AGND (Volts)	ADC0H:ADC0L (AD0LJST=0) <i>Right Justified</i>	ADC0H:ADC0L (AD0LJST=1) <i>Left Justified</i>
$VREF \times \frac{4095}{4096}$	0FFFH	FFF0H
$\frac{VREF}{2}$	0800H	8000H
$VREF \times \frac{2047}{4096}$	07FFH	7FF0H
0	0000H	0000H



◆ ניתן לתכנן ממיר ADC0 לפי רצף הבא:

- צעד 1: קובעים מתח ייחוס. (REF0CN)
- צעד 2: קובעים קצב ההמרה (תדר השעון SAR0) והגבר של מערכת (PGA0) באוגר ADC0CF.
- צעד 3: מגדירים ערוצים של מרבב אנלוגי בכניסה. (AMX0CF)
- צעד 4: בוחרים את הערוץ הספציפי שנעבוד איתו. (AMX0SL)
- צעד 5: מדליקים סיביות בקרה ומתחילים תהליך המרה לאחר הדלקת ADC0CN.



2.4V Output of Internal VREF

Bit	Symbol	Description
7-5	-	Unused. Read=000b; Write=Don't care.
4	AD0VRS	ADC0 Voltage Reference Select 0: ADC0 voltage reference from VREF0 pin. 1: ADC0 voltage reference from DAC0 output.
3	AD1VRS	ADC1 Voltage Reference Select 0: ADC1 voltage reference from VREF1 pin. 1: ADC1 voltage reference from AV+
2	TEMPE	Temperature Sensor Enable Bit 0: Internal Temperature Sensor Off. 1: Internal Temperature Sensor On.
1	BIASE	ADC/DAC Bias Generator Enable Bit. (Must be '1' if using ADC or DAC) 0: Internal Bias Generator Off. 1: Internal Bias Generator On.
0	REFBE	Internal Reference Buffer Enable Bit. 0: Internal Reference Buffer Off. 1: Internal Reference Buffer On. Internal voltage reference is driven on the VREF pin.

Bit	Symbol	Description
7-3	AD0SC4-0	<p>ADC0 SAR0 Conversion Clock frequency Bits SAR0 Conversion clock is derived from system clock by the following equation, where AD0SC refers to the 5-bit value in AD0SC4-0 and CLK_{SAR0} refers to the desired ADC0 SAR conversion clock frequency.</p> $AD0SC = \frac{SYSCLK}{CLK_{SAR0}} - 1$
2-0	AMP0GN2-0	<p>ADC0 Internal Amplifier Gain (PGA) 000: Gain = 1 001: Gain = 2 010: Gain = 4 011: Gain = 8 10x: Gain = 16 11x: Gain = 0.5</p>

קביעת תדר של שעון למטרת ההמרה בעזרת אוגר SAR0

- ◆ תדר המרה המקסימלי הוא 2.5 MHz
- ◆ ניתן לחשב תדר של הממיר בעזרת נוסחה הבאה:

$$CLK_{SAR0} = \frac{SYSCLK}{AD0SC + 1}$$

- ◆ לדוגמא: אם תדר השעון המערכת הוא ב-16 MHz וערך של AD0SC4-0 שווה ל-10000b, אז תדר ההמרה של SAR0 שווה ל- $16\text{MHz}/17 = 941.176\text{ KHz}$.
- ◆ נניח שהכנסנו ערך 10000000 לתוך אוגר **ADC0CF**, תדר ההמרה של SAR0 יהיה 941 KHz והגבר PGA0 יהיה 1.



במידה וסיבית שווה ל-0 ישנם 2 כניסות חד-קוטביות, 1 – כניסה הפרשית דרך 2 פינים.

Bit	Symbol	Description
7-4	-	UNUSED. Read=0000, Write=don't care
3	AIN67IC	<i>AIN6, AIN7 Input Pair Configuration Bit</i> 0: AIN6 and AIN7 are independent single-ended inputs 1: AIN6, AIN7 are (respectively) +,- differential input pair
2	AIN45IC	<i>AIN4, AIN5 Input Pair Configuration Bit</i> 0: AIN4 and AIN5 are independent single-ended inputs 1: AIN4, AIN5 are (respectively) +,- differential input pair
1	AIN23IC	<i>AIN2, AIN3 Input Pair Configuration Bit</i> 0: AIN2 and AIN3 are independent single-ended inputs 1: AIN2, AIN3 are (respectively) +,- differential input pair
0	AIN01IC	<i>AIN0, AIN1 Input Pair Configuration Bit</i> 0: AIN0 and AIN1 are independent single-ended inputs 1: AIN0, AIN1 are (respectively) +,- differential input pair

AMX0SL—AMUX0 אוגר לבחירת ערוץ של מרבב אנלוגי

Bit	Symbol	Description
7-4	-	UNUSED. Read=0000, Write=don't care
3-0	AMX0AD3-0	AMX0 Address Bits 0000-1111 טבלת ההמרה רשומה בשקף בעמוד הבא

AMUX0 Channel Selection—AMX0SL SFR

		AMX0AD3-0								
		0000	0001	0010	0011	0100	0101	0110	0111	1xxx
AMX0CF Bits 3-0	0000	AIN0	AIN1	AIN2	AIN3	AIN4	AIN5	AIN6	AIN7	TEMP SENSOR
	0001	+(AIN0) -(AIN1)		AIN2	AIN3	AIN4	AIN5	AIN6	AIN7	TEMP SENSOR
	0010	AIN0	AIN1	+(AIN2) -(AIN3)		AIN4	AIN5	AIN6	AIN7	TEMP SENSOR
	0011	+(AIN0) -(AIN1)		+(AIN2) -(AIN3)		AIN4	AIN5	AIN6	AIN7	TEMP SENSOR
	0100	AIN0	AIN1	AIN2	AIN3	+(AIN4) -(AIN5)		AIN6	AIN7	TEMP SENSOR
	0101	+(AIN0) -(AIN1)		AIN2	AIN3	+(AIN4) -(AIN5)		AIN6	AIN7	TEMP SENSOR
	0110	AIN0	AIN1	+(AIN2) -(AIN3)		+(AIN4) -(AIN5)		AIN6	AIN7	TEMP SENSOR
	0111	+(AIN0) -(AIN1)		+(AIN2) -(AIN3)		+(AIN4) -(AIN5)		AIN6	AIN7	TEMP SENSOR
	1000	AIN0	AIN1	AIN2	AIN3	AIN4	AIN5	+(AIN6) -(AIN7)		TEMP SENSOR
	1001	+(AIN0) -(AIN1)		AIN2	AIN3	AIN4	AIN5	+(AIN6) -(AIN7)		TEMP SENSOR
	1010	AIN0	AIN1	+(AIN2) -(AIN3)		AIN4	AIN5	+(AIN6) -(AIN7)		TEMP SENSOR
	1011	+(AIN0) -(AIN1)		+(AIN2) -(AIN3)		AIN4	AIN5	+(AIN6) -(AIN7)		TEMP SENSOR
	1100	AIN0	AIN1	AIN2	AIN3	+(AIN4) -(AIN5)		+(AIN6) -(AIN7)		TEMP SENSOR
	1101	+(AIN0) -(AIN1)		AIN2	AIN3	+(AIN4) -(AIN5)		+(AIN6) -(AIN7)		TEMP SENSOR
1110	AIN0	AIN1	+(AIN2) -(AIN3)		+(AIN4) -(AIN5)		+(AIN6) -(AIN7)		TEMP SENSOR	
1111	+(AIN0) -(AIN1)		+(AIN2) -(AIN3)		+(AIN4) -(AIN5)		+(AIN6) -(AIN7)		TEMP SENSOR	



ADC0CN—ADC0 אוגר בקרה

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
AD0EN	AD0TM	AD0INT	AD0BUSY	AD0CM1	AD0CM0	AD0WINT	AD0LJST	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
						(bit addressable)		0xE8

Bit7: AD0EN: ADC0 Enable Bit.
 0: ADC0 Disabled. ADC0 is in low-power shutdown.
 1: ADC0 Enabled. ADC0 is active and ready for data conversions.

Bit6: AD0TM: ADC Track Mode Bit
 0: When the ADC is enabled, tracking is continuous unless a conversion is in process
 1: Tracking Defined by ADSTM1-0 bits

Bit5: AD0INT: ADC0 Conversion Complete Interrupt Flag.
 This flag must be cleared by software.
 0: ADC0 has not completed a data conversion since the last time this flag was cleared.
 1: ADC0 has completed a data conversion.

Bit4: AD0BUSY: ADC0 Busy Bit.
 Read:
 0: ADC0 Conversion is complete or a conversion is not currently in progress. AD0INT is set to logic 1 on the falling edge of AD0BUSY.
 1: ADC0 Conversion is in progress.
 Write:
 0: No Effect.
 1: Initiates ADC0 Conversion if AD0STM1-0 = 00b

Bit3-2: AD0CM1-0: ADC0 Start of Conversion Mode Select.
 If AD0TM = 0:
 00: ADC0 conversion initiated on every write of '1' to AD0BUSY.
 01: ADC0 conversion initiated on overflow of Timer 3.
 10: ADC0 conversion initiated on rising edge of external CNVSTR.
 11: ADC0 conversion initiated on overflow of Timer 2.
 If AD0TM = 1:
 00: Tracking starts with the write of '1' to AD0BUSY and lasts for 3 SAR clocks, followed by conversion.
 01: Tracking started by the overflow of Timer 3 and last for 3 SAR clocks, followed by conversion.
 10: ADC0 tracks only when CNVSTR input is logic low; conversion starts on rising CNVSTR edge.
 11: Tracking started by the overflow of Timer 2 and last for 3 SAR clocks, followed by conversion.

Bit1: AD0WINT: ADC0 Window Compare Interrupt Flag.
 This bit must be cleared by software.
 0: ADC0 Window Comparison Data match has not occurred since this flag was last cleared.
 1: ADC0 Window Comparison Data match has occurred.

Bit0: AD0LJST: ADC0 Left Justify Select.
 0: Data in ADC0H:ADC0L registers are right-justified.
 1: Data in ADC0H:ADC0L registers are left-justified.



◆ שיטת Polling :

- ניתן לבדוק סיבית AD0INT (ADC0CN.5) כדי לבדוק האם המרה הסתיימה.
bit may be polled to determine when a conversion has completed
- ברגע שסיבית שווה ל-"1" ניתן לקרוא נתונים מ-ADC0

◆ שיטת פסיקות:

- במידה ופסיקות מאופשרות (גם ADC0 End-of-Conversion (EIE2.1) וגם פסיקות גלובליות), אז ברגע שהמרה תסתיים נקבל פסיקה ושגרה ADC0 ISR תתבצע.
- ניתן לקרוא נתונים מ-ADC0 בתוך תת שגרה ADC0 ISR.

Polling Method דוגמא ADC0

```
void Init_ADC0(void)
{
    REF0CN = 0x07;    //--Enable internal bias generator and
                    // internal reference buffer
                    // Select ADC0 reference from VREF0 pin
                    // Internal Temperature Sensor ON
    ADC0CF = 0x81;    //--SAR0 conversion clock=941KHz approx
                    // Gain=2
    AMX0SL = 0x08;    //--Select Temp Sensor
    ADC0CN = 0x80;    //--Enable ADC0, Continuous Tracking
                    // Mode Conversion initiated on write to
                    // AD0BUSY; ADC0 data is right justified.
}

void main (void)
{
    Device_Init ();    // Init device peripherals

    AD0BUSY = 1;      // Start ADC conversion
    while (!AD0INT);  // Wait till conversion is complete
    ADC0_Value = ADC0; // Store ADC result in variable
    AD0INT = 0;       // Clear AD0INT flag

    while (1);        // Spin forever
}
```



SILICON LABS

- ◆ משתמשים בטיימר 3 כדי להפעיל המרה בעזרת ADC0.
- ◆ מאפשרים פסיקה של טיימר 3 (לא כתוב בקוד הנוכחי).
- ◆ תת שגרה של טיימר 3 (Timer 3 ISR) מופעלת כאשר מתחיל תהליך המרה של ADC.
- ◆ תוך כדי תת שגרה של טיימר 3 פועלת אנו קודם כל מאפסים דגל **TF3** (דגל מילוי יתר) ואז בודקים את הדגל **AD0INT**. ברגע שהוא עולה ל-1 עוברים לשלב הבא.
- ◆ ברגע שדגל **AD0INT** עלה ל- "1", אנו מבינים שתהליך ההמרה הסתיים.
- ◆ בסיום התהליך אנו קוראים מידע מאוגר ADC0 ושומרים אותו במשתנה *ADC0_reading*.

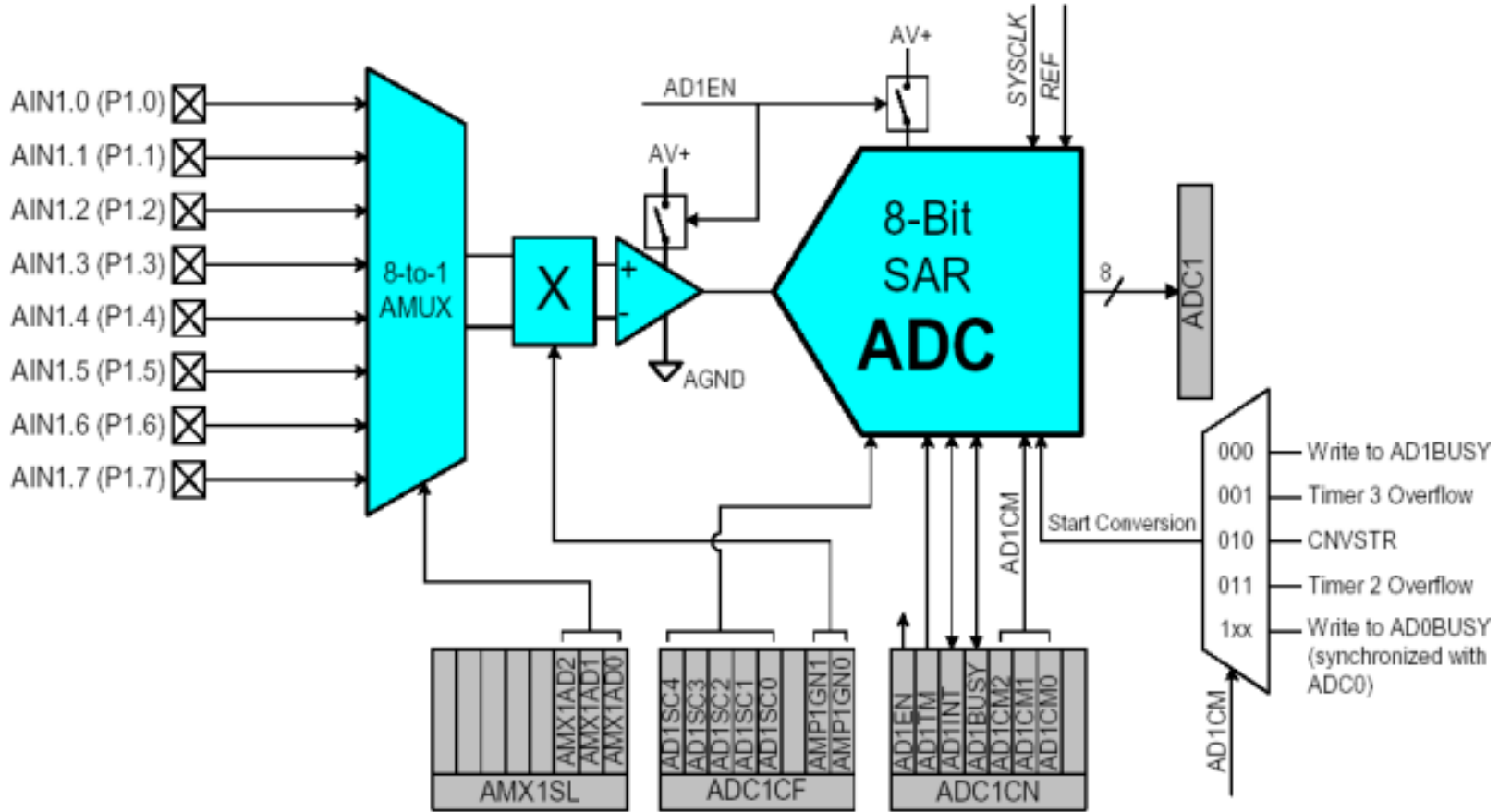
- ◆ אנו יכולים להשתמש בפסיקה של ADC0 לאחר אפשר פסיקות באופן כללי, ופסיקת ההמרה (EIE2.1) EADC0 בפרט.
- ◆ כל פעם שהמרה תסתיים תוכנה תקרא לתת שגרה ISR ADC0.
- ◆ בתוך ISR אנו צריכים לעשות פעולות הבאות:
 - לקרוא את הערך שנמצא בתוך אוגר ADC0.
 - לשמור את הערך בתוך משתנה עזר (או מערך).
 - לנקות ערך של דגל AD0INT.


```

void Init_ADC0(void)
{
    REF0CN = 0x07;          //-- Enable internal bias generator and
                            // internal reference buffer
                            // Select ADC0 reference from VREF0 pin
                            // Internal Temperature Sensor ON
    ADC0CF = 0x81;          //-- SAR0 conversion clock=941KHz approx
                            // Gain=2
    AMX0SL = 0x08;          //-- Select Temp Sensor
    ADC0CN = 0x84;          //-- Enable ADC0, Continuous Tracking
                            // Mode, Conversion initiated on Timer
                            // 3 overflow, ADC0 data is right
                            // justified

    EIE2 |= 0x02;          //-- Enable ADC Interrupts
}
//-----
void ADC0_ISR (void) interrupt 15
{
    AD0INT = 0;             //-- Clear ADC0 conversion complete
                            // interrupt flag
    ADC0_reading = ADC0;    //-- Read ADC0 data
}
    
```

8-Bit ADC (ADC1)



8-Bit ADC (ADC1)

◆ מערכת ADC1 בנויה ממודולים הבאים:

➤ מרבב אנלוגי 8 ערוצים (AMUX1)

➤ מגבר עם הגבר ניתן לשינוי (PGA1).

▪ הגבר ברירת מחדל 0.5 .

▪ ניתן לשנות הגבר לאחד מערכים הבאים: 0.5, 1, 2, 4 .

➤ ממיר ADC 8 סיביות.

◆ ניתן להפעיל ADC1 ע"י עליית סיבית (ADC1CN.7) AD1EN ל-"1".

תחילת ההמרה של ADC1.

◆ ניתן להתחיל המרה באחד מחמש דרכים שונות. תלוי במצבים של סיביות **AD1CM2-0 של אוגר ADC1CN.**

1. פקודה ברמת תוכנה ("1" ב-AD1BUSY). מפעיל ביחד עם ADC0

2. מילוי יתר של טיימר 2.

3. מילוי יתר של טיימר 3.

4. אות חיצוני (פולס עלייה ב- CNVSTR).

5. כתיבת "1" ל- **AD0BUSY (ADC0CN.4)**

◆ סיבית **AD1BUSY** נשאר במצב "1" במשך זמן ההמרה והופך ל-"0", כאשר המרה מסתיימת.

◆ פולס ירידה של **AD1BUSY** מפעיל פסיקה (אם היא מאופשרת) ומעלה ל-"1" דגל פסיקה **AD1INT**.

◆ במידה ופסיקה של "סיום ההמרה" מאופשרת, אז תהיה פסיקה, כאשר **AD1INT** יהיה במצב "1" ואז תת-שגרה ADC1 ISR תתחיל להתבצע.



◆ מיפוי של ADC1 הרבה יותר פשוט מ-ADC0 כי יש רק אוגר אחד שמכיל ערך של מילה לאחר ההמרה.

◆ אין צורך להצמיד ערכים לצד שמאל או לצד ימין.

◆ נוסחה לחישוב קוד לאחר ההמרה:

$$ADC1Code = Vin \times \frac{Gain}{VREF} \times 256$$

AIN1.0 – AGND (Volts)	ADC1
$VREF \times \frac{255}{256}$	FFH
$\frac{VREF}{2}$	80H
$VREF \times \frac{127}{256}$	7FH
0	00H

◆ כדי לתכנן ADC1 יש לבצע צעדים הבאים:

➤ צעד 1: קובעים מתח ייחוס . (REF0CN)

➤ צעד 2: מגדירים פינים מיוחדים בפורט 1 בתור פורטים אנלוגיים. (P1MDIN)

➤ מגדירים תדר המרה SAR1 והגבר מתח PGA1 . (ADC1CF)

➤ בוחרים ערוץ של מרבב אנלוגי. (AMX1SL).

➤ מגדירים סיביות בקרה ומפעילים ממיר. (ADC1CN)

Bit	Symbol	Description
7-3	AD1SC4-0	<p>ADC1 SAR Conversion Clock frequency Bits SAR Conversion clock is derived from system clock by the following equation, where AD1SC refers to the 5-bit value in AD1SC4-0, and CLK_{SAR1} refers to the desired ADC1 SAR conversion clock frequency.</p> $AD1SC = \frac{SYSCLK}{CLK_{SAR1}} - 1$
2	-	UNUSED. Read=0, Write=don't care
1-0	AMP1GN1-0	<p>ADC1 Internal Amplifier Gain (PGA) 00: Gain = 0.5 01: Gain = 1 10: Gain = 2 11: Gain = 4</p>

◆ תדר המרה מקסימלי 6 MHz

◆ ניתן לחשב תדר ההמרה בעזרת נוסחה הבאה:

$$CLK_{SAR1} = \frac{SYSCLK}{AD1SC + 1}$$

אוגר לבחירת ערוץ של מרבב אנלוגי AMX1SL—AMUX1

Bit	Symbol	Description
7-3	-	UNUSED. Read=00000, Write=don't care
3-0	AMX1AD2-0	AMX1 Address Bits 000: AIN1.0 selected 001: AIN1.1 selected 010: AIN1.2 selected 011: AIN1.3 selected 100: AIN1.4 selected 101: AIN1.5 selected 110: AIN1.6 selected 111: AIN1.7 selected

ADC1CN—ADC1 אוגר בקרה

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
AD1EN	AD1TM	AD1INT	AD1BUSY	AD1CM2	AD1CM1	AD1CM0	-	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xAA
Bit7:	AD1EN: ADC1 Enable Bit. 0: ADC1 Disabled. ADC1 is in low-power shutdown. 1: ADC1 Enabled. ADC1 is active and ready for data conversions.							
Bit6:	AD1TM: ADC1 Track Mode Bit. 0: Normal Track Mode: When ADC1 is enabled, tracking is continuous unless a conversion is in process. 1: Low-power Track Mode: Tracking Defined by AD1STM2-0 bits (see below).							
Bit5:	AD1INT: ADC1 Conversion Complete Interrupt Flag. This flag must be cleared by software. 0: ADC1 has not completed a data conversion since the last time this flag was cleared. 1: ADC1 has completed a data conversion.							
Bit4:	AD1BUSY: ADC1 Busy Bit. Read: 0: ADC1 Conversion is complete or a conversion is not currently in progress. AD1INT is set to logic 1 on the falling edge of AD1BUSY. 1: ADC1 Conversion is in progress. Write: 0: No Effect. 1: Initiates ADC1 Conversion if AD1STM2-0 = 000b							
Bit3-1:	AD1CM2-0: ADC1 Start of Conversion Mode Select. AD1TM = 0: 000: ADC1 conversion initiated on every write of '1' to AD1BUSY. 001: ADC1 conversion initiated on overflow of Timer 3. 010: ADC1 conversion initiated on rising edge of external CNVSTR. 011: ADC1 conversion initiated on overflow of Timer 2. 1xx: ADC1 conversion initiated on write of '1' to AD0BUSY (synchronized with ADC0 software-commanded conversions). AD1TM = 1: 000: Tracking initiated on write of '1' to AD1BUSY and lasts 3 SAR1 clocks, followed by conversion. 001: Tracking initiated on overflow of Timer 3 and lasts 3 SAR1 clocks, followed by conversion. 010: ADC1 tracks only when CNVSTR input is logic low; conversion starts on rising CNVSTR edge. 011: Tracking initiated on overflow of Timer 2 and lasts 3 SAR1 clocks, followed by conversion. 1xx: Tracking initiated on write of '1' to AD0BUSY and lasts 3 SAR1 clocks, followed by conversion.							
Bit0:	UNUSED. Read = 0b. Write = don't care.							



SILICON LABS

◆ שיטת Polling

➤ ניתן לבדוק סיבית AD1INT (ADC1CN.5) כדי לבדוק האם המרה הססתימה או לא.

➤ ברגע שסיבית עלתה ל-"1" יש לבדוק אוגר של ADC1.

◆ שיטת פסיקות

➤ במידה ופסיקות גלובליות מאופשרות ופסיקת סיום ההמרה מאופשרת (EIE2.3), אז תת שגרה של פסיקת ההמרה ADC1 ISR מתבצעת.

➤ בתוך תת שגרה ADC1 ISR, קוראים מילה מאוגר ADC1.



דוגמא לתכנון ADC1 בעזרת שיטת Polling

```
void Init_ADC1(void)
{
    REF0CN = 0x03;          //-- Enable internal bias generator and
                            // internal reference buffer
                            // Select ADC1 reference from VREF1 pin
    ADC1CF = 0x81;          //-- SAR1 conversion clock=941KHz approx., Gain=1
    AMX1SL = 0x00;          //-- Select AIN1.0 input
    ADC1CN = 0x82;          //-- Enable ADC1, Continuous Tracking Mode,
                            // Conversion initiated on Timer 3 overflow
}
//-----
// Interrupt Service Routine
void Timer3_ISR (void) interrupt 14
{
    TMR3CN &= ~(0x80);      //-- Clear TF3 flag
                            //-- Wait for ADC1 conversion to be over
    while ( (ADC1CN &= 0x20) == 0); //-- Poll for AD1INT-->1
    ADC1_reading = ADC1;    //-- Read ADC1 data
    ADC1CN &= 0xDF;        //-- Clear AD1INT
}
```



דוגמא לתכנון ADC1 בעזרת שיטת פסיקות.

- ◆ במקום שימוש בשיטת Polling ניתן להשתמש בשיטת פסיקות.
- ◆ מאפשרים פסיקה של ממיר ADC1 ע"י אפשור של סיבית EADC1 (EIE2.3) ואפשור גלובלי של פסיקות.
- ◆ תת שגרה ISR של ADC1 תעבוד מחדש כל פעם שהמרה תסתיים.
- ◆ בתוך תת שגרה אנו צריכים:
 - לקרוא לאוגר ADC1.
 - לשמור ערך לתוך משתנה.
 - לנקות דגל AD1INT





S I L I C O N L A B S

www.silabs.com/MCU