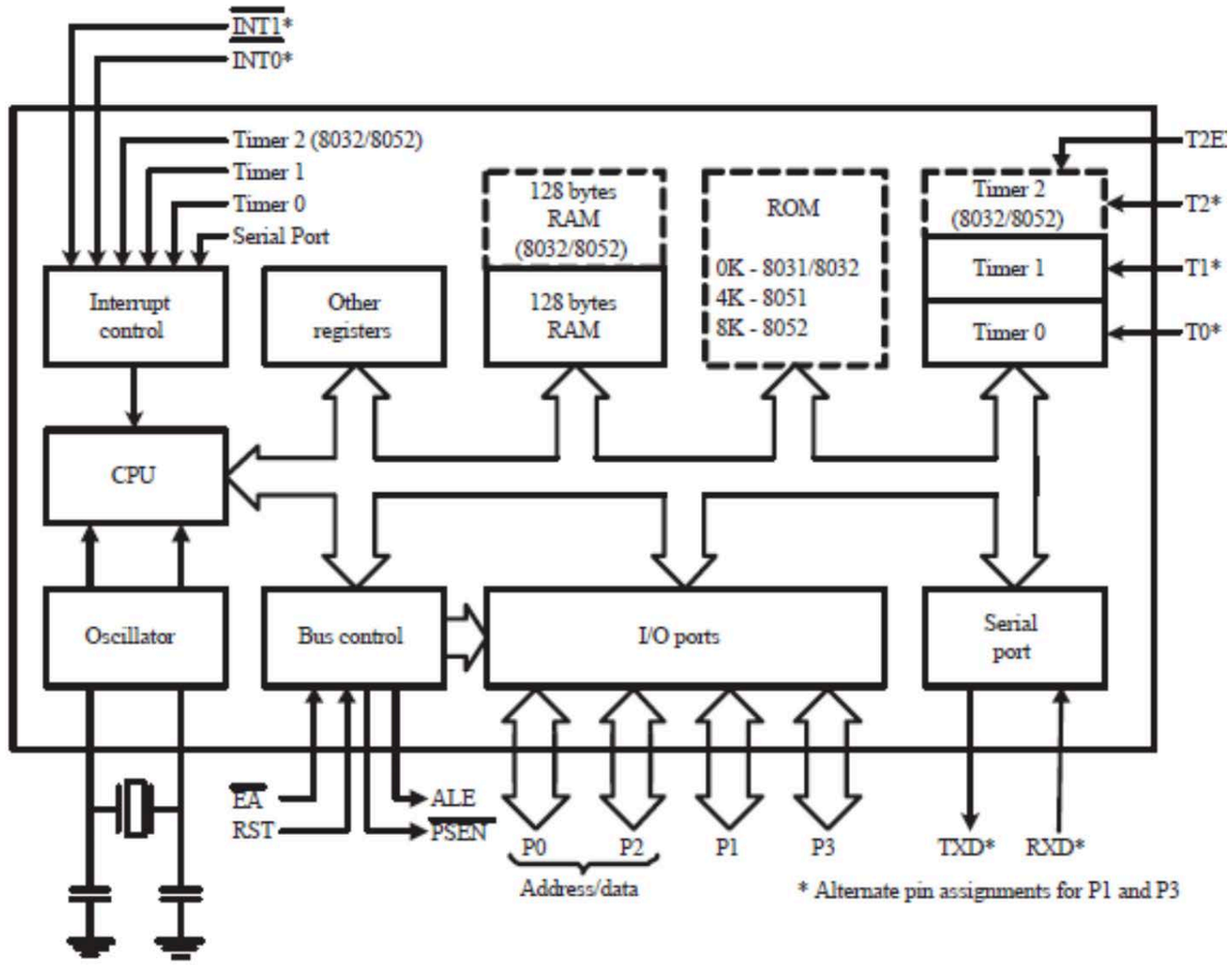
ECEN202

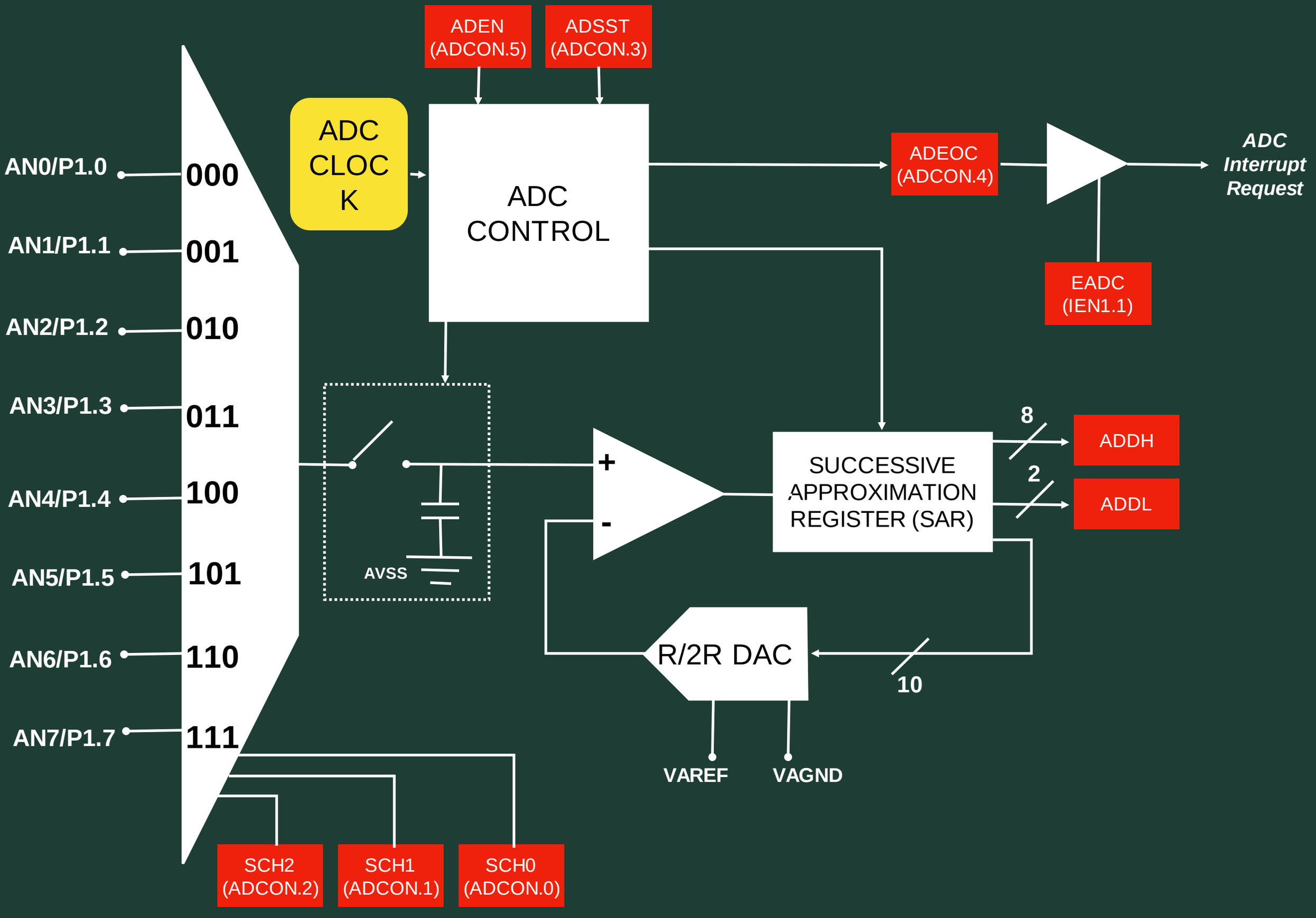


Mohammad Nekooei
mohammad.nekooei@ecsvuw.ac.nz

THIS WEEK

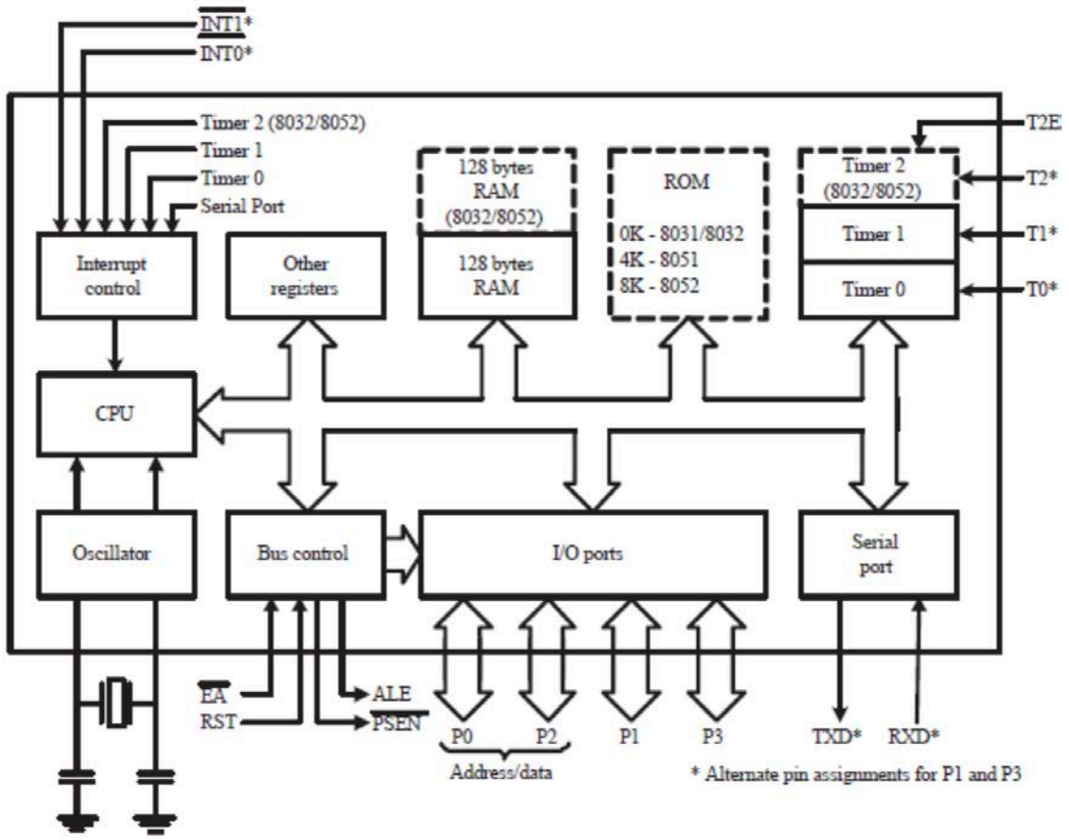
- Today:
  - Finish ADC discussion.
    - ADC terminology
    - ADC modes on the AT89C51AC3
    - ADC interrupts
    - Example code

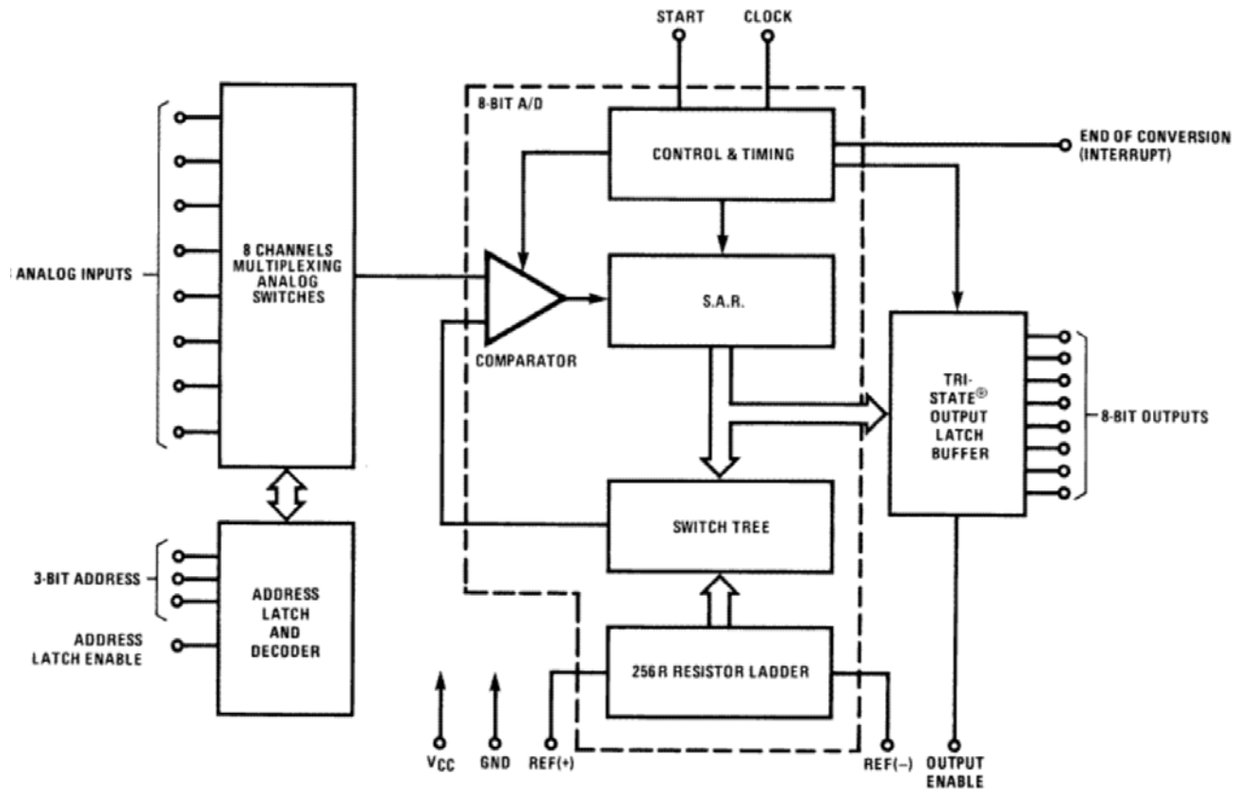
ADC DIAGRAM


ADEN (ADCON.5)
ADSST (ADCON.3)
ADC CLOCK
ADC CONTROL
ADEOC (ADCON.4)
ADC Interrupt Request
EADC (IEN1.1)
AN0/P1.0
AN1/P1.1
AN2/P1.2
AN3/P1.3
AN4/P1.4
AN5/P1.5
AN6/P1.6
AN7/P1.7
000
001
010
011
100
101
110
111
AVSS
+
-
SUCCESSIVE APPROXIMATION REGISTER (SAR)
8
2
ADDH
ADDL
R/2R DAC
10
VAREF
VAGND
SCH2 (ADCON.2)
SCH1 (ADCON.1)
SCH0 (ADCON.0)

ANALOGUE TO DIGITAL

- The original Intel 8051 did not have an on-board Analogue to Digital Converter (ADC). This was due to cost and size constraints at the time.
  - If applications needed an ADC, external ADCs were used. These were generally interfaced with in a parallel manner. An 8-bit converter would be connected to all eight pins of Port 1, for example.
    - The Texas Instruments ADC0808 is one example of a multichannel 8-bit parallel ADC of the type that might be used with non-ADC 8051 variants.



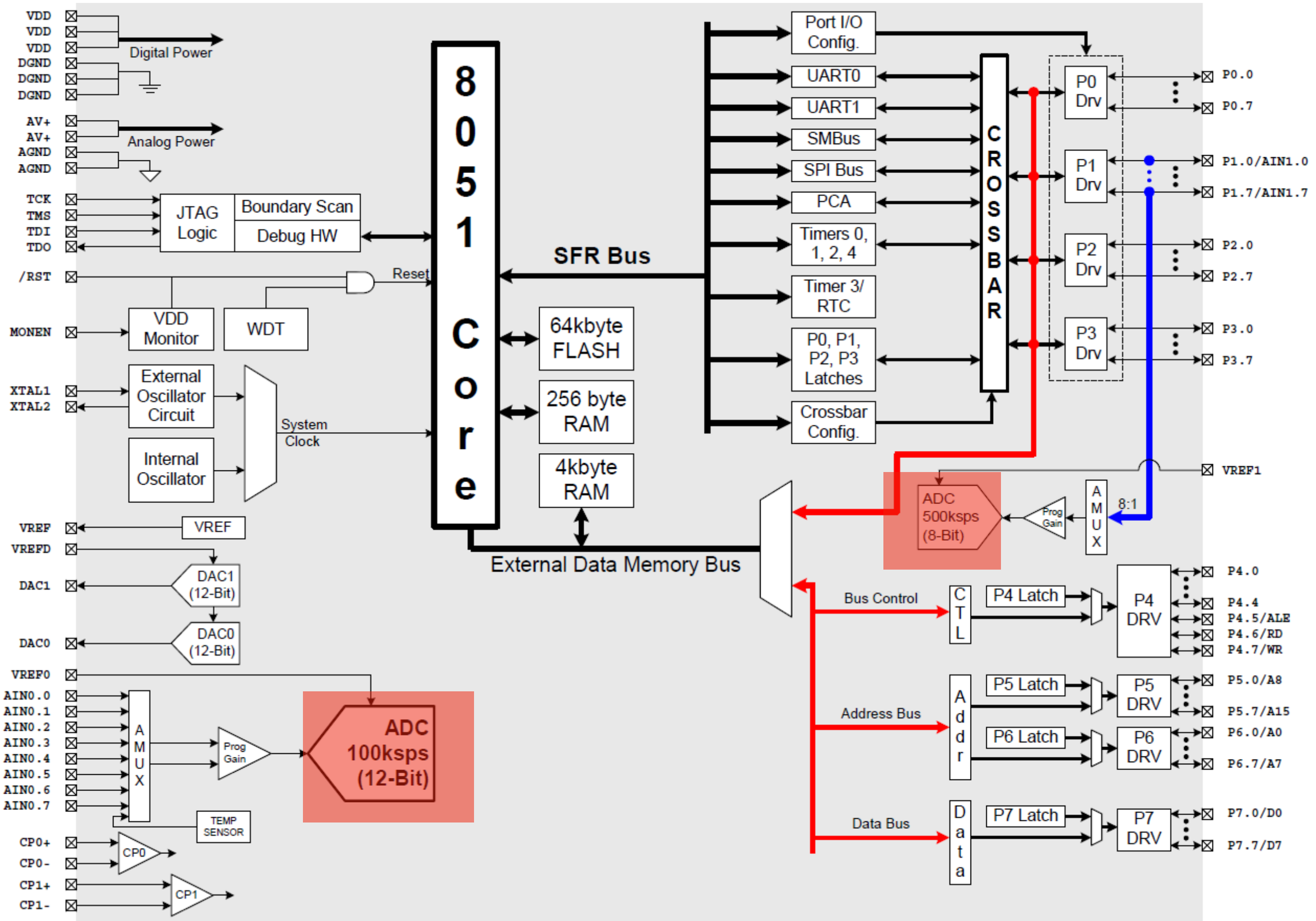


C8051F020x ADC

- It is very common for contemporary binary-compatible 8051 variants to have onboard ADC peripherals.
  - This follows the trend of increasing integration in microcontrollers.
- The C8051F020x has an onboard ADC with up to 12-bit resolution.
  - 2^12 = 4096 values

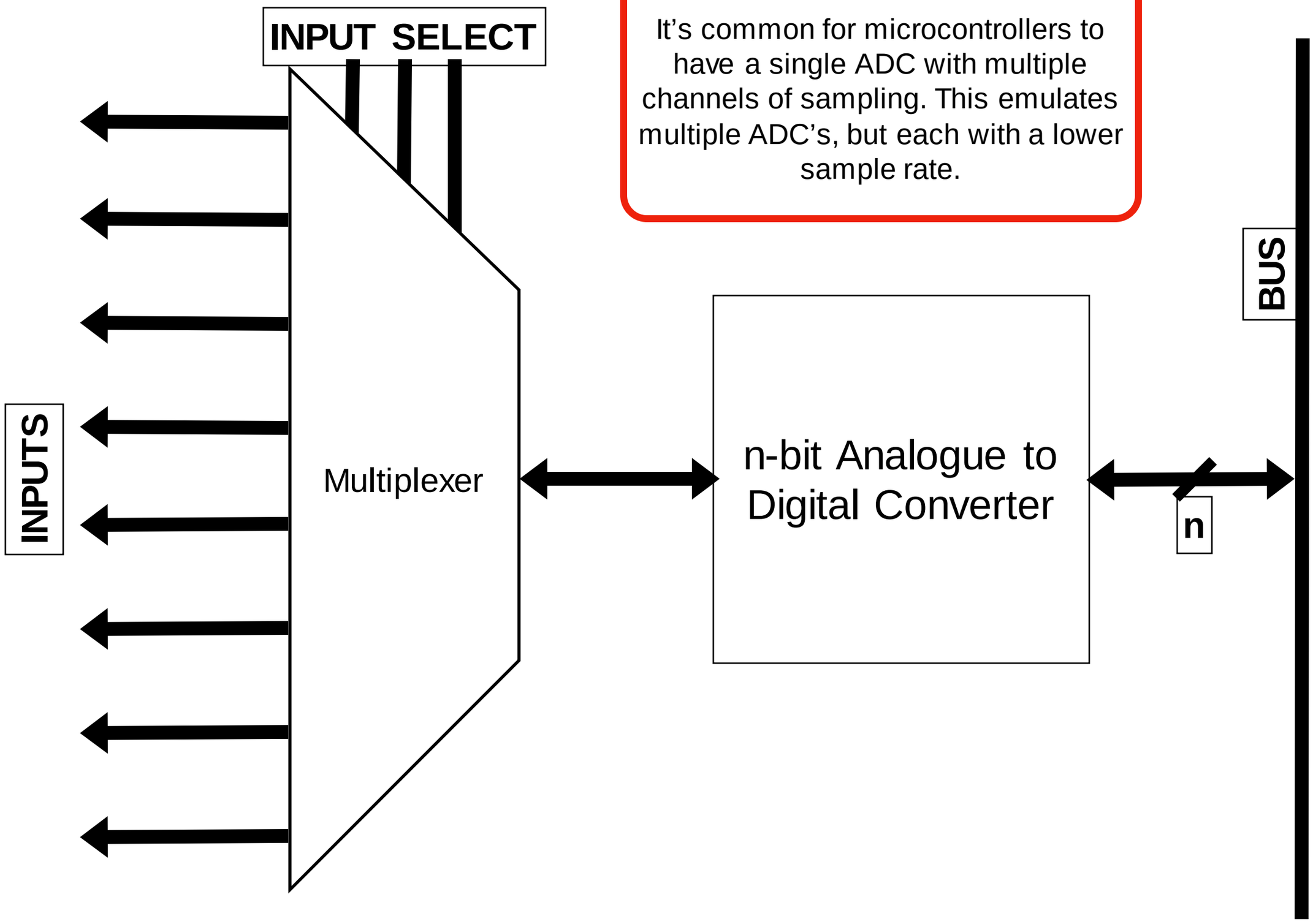
- Two on-board ADC (ADC0, ADC1)
- The onboard 8-bit and 12-bit ADCs are single converters but are multiplexed.
  - This multiplexing allows for 8 separate inputs.

C8051F020x ADC

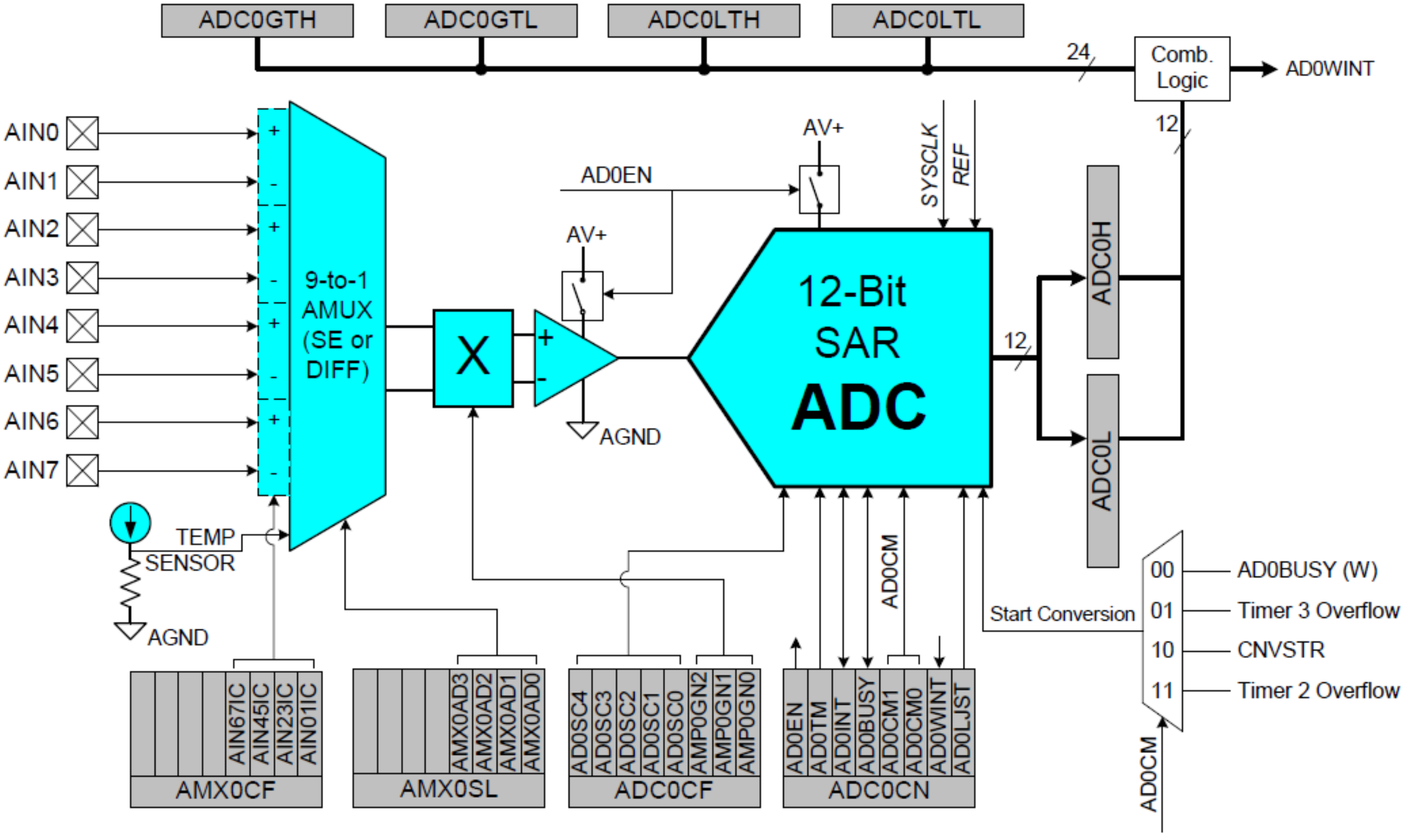


MULTICHANNEL SAMPLING

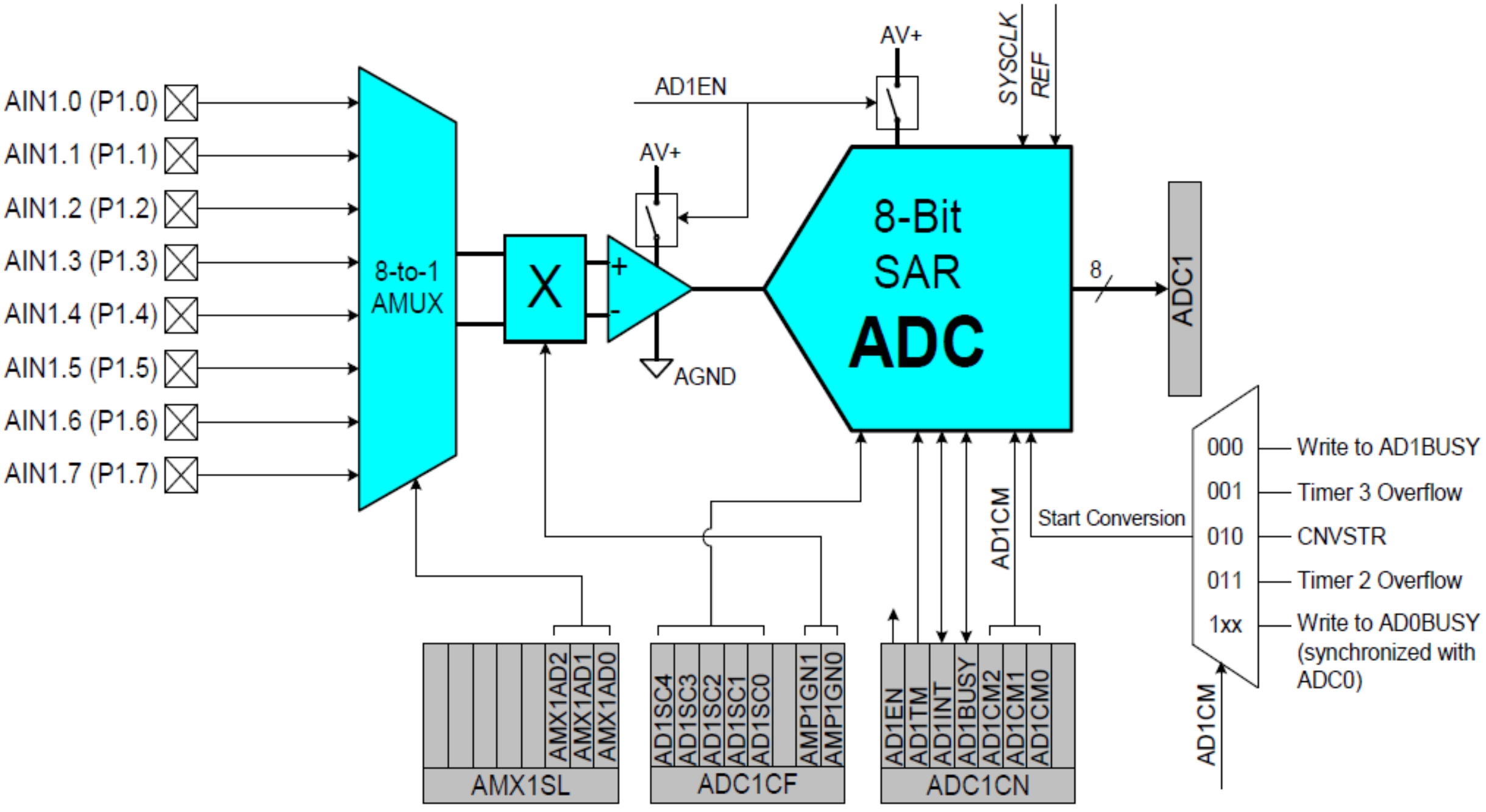
It’s common for microcontrollers to have a single ADC with multiple channels of sampling. This emulates multiple ADC’s, but each with a lower sample rate.



C8051F020x 12-Bit ADC0 DIAGRAM



C8051F020x 8-Bit ADC1 DIAGRAM

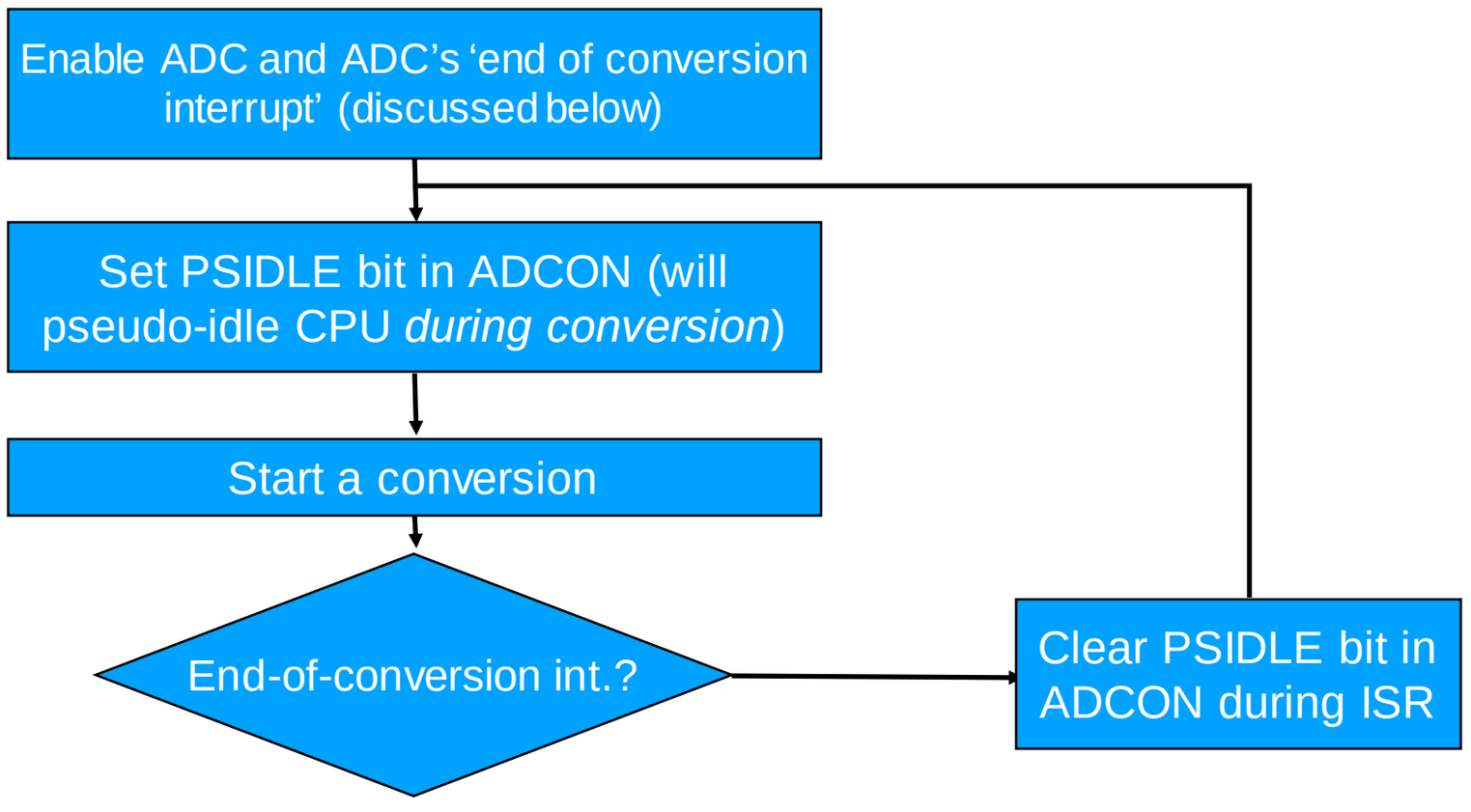


ADC TERMINOLOGY

- External VREF (reference voltage) range: 2.4 Volts, typically
  - The external reference voltage against which the ADC input is compared.
- ADCIN voltage range: 0 V to VREF V
  - Input signals must be scaled to not exceed 3.0 V
- Non-linearity: max ±1 LSB
  - “the deviation of the transfer function from a straight line”
    - In the case of the C8051F02x, this will be a line no more than 1 LSB off of the ideal line.
  - See https://www.maximintegrated.com/en/app-notes/index.mvp/id/641 for a detailed glossary of ADC terminology.

8-BIT & 12-BIT MODES

- The C8051F02x's ADC has two modes: an 8-bit mode and a higher-resolution "precision" 12-bit mode.
- The AT89C51AC3's ADC has two modes: an 8-bit mode and a higher-resolution "precision" 10-bit mode.
  - To take advantage this precision mode, the CPU must be put into a 'pseudo-idle' mode.
    - This is done to reduce digital noise generated by the CPU. Too much CPU noise will corrupt the higher-resolution 10-bit conversion.
    - In pseudo-idle mode (PSIDLE), peripherals continue to run.



12-Bit ADC (ADC0)

- The ADC0 subsystem consists of:
  - 9-channel, configurable analog multiplexer (AMUX0)
    - 8 channels for external input
      - Single-ended inputs
      - Differential input pairs
    - 9th channel for on-chip temperature measurement
  - Programmable gain amplifier (PGA0)
    - Default gain is 1
    - Gain can be programmed to be 0.5, 1, 2, 4, 8 or 16
  - 12-bit Successive approximation register (SAR) ADC

- ADC0 is enabled by setting **AD0EN** (**ADC0CN.7**) to 1

Starting ADC0 Conversions

- Conversions can be started in four different ways (depending on the **AD0CM1** and **AD0CM0** bits in **ADC0CN** register)
  1. Software command (writing 1 to **AD0BUSY**)
  2. Overflow of timer 2
  3. Overflow of timer 3
  4. External signal input (rising edge of **CNVSTR**)

- The **AD0BUSY** bit remains set to 1 during conversion and restored to 0 when the conversion is complete

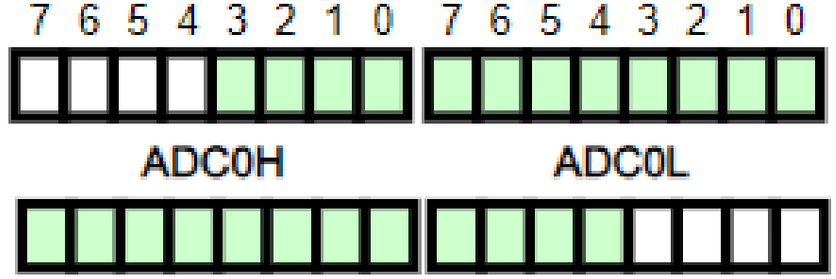
- The falling edge of **AD0BUSY** triggers an interrupt (when enabled) and sets the **AD0INT** interrupt flag (**ADC0CN.5**)

- If ADC0 end-of-conversion interrupt (**EIE2.1**) is enabled, then an interrupt will be generated when **AD0INT** is set and the appropriate ADC0 ISR will be executed

SILICON LABS

Data Word Conversion Map (12-bit)

- Converted data is stored in the **ADC0H** and **ADC0L** registers and can be either left- or right-justified in the register pair depending on the programmed state of the **AD0LJST** (**ADC0CN.0**) bit
  - 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)



- The mapping of the ADC0 analog inputs to the ADC0 data word registers is given by:

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

where n=12 for single-ended and n=11 for differential inputs

Data Word Conversion Map (12-bit)

- Suppose AIN0 is used as the input in single-ended mode (**AMX0CF=00H** and **AMXSL=00H**) and gain is set to 1

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

Programming ADC0

- ADC0 can be programmed through the following sequence
  - Step 1: configure the voltage reference (REF0CN)
  - Step 2: set the SAR0 conversion clock frequency and PGA0 gain (ADC0CF)
  - Step 3: configure the multiplexer input channels (AMX0CF)
  - Step 4: select the desired multiplexer input channel (AMX0SL)
  - Step 5: set the appropriate control bits and start-of-conversion mode and turn on ADC0 (ADC0CN)

Reference Control Register (REF0CN)

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.

Configuration Register (ADC0CF)

Bit	Symbol	Description
7-3	**AD0SC4-0**	***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. $$AD0SC = \frac{SYSCLK}{CLK_{SAR0}} - 1$$
2-0	**AMP0GN2-0**	***ADC0 Internal Amplifier Gain (PGA)*** 000: Gain = 1 001: Gain = 2 010: Gain = 4 011: Gain = 8 10x: Gain = 16 11x: Gain = 0.5

SAR0 Conversion Clock Frequency

- The conversion clock has a maximum frequency of **2.5 MHz**
- The conversion clock frequency is calculated using the following equation:

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

- If the System Clock Frequency is 16 MHz and AD0SC4-0 is set to 10000b, then the SAR0 conversion frequency is 16MHz/17 = 941.176 KHz

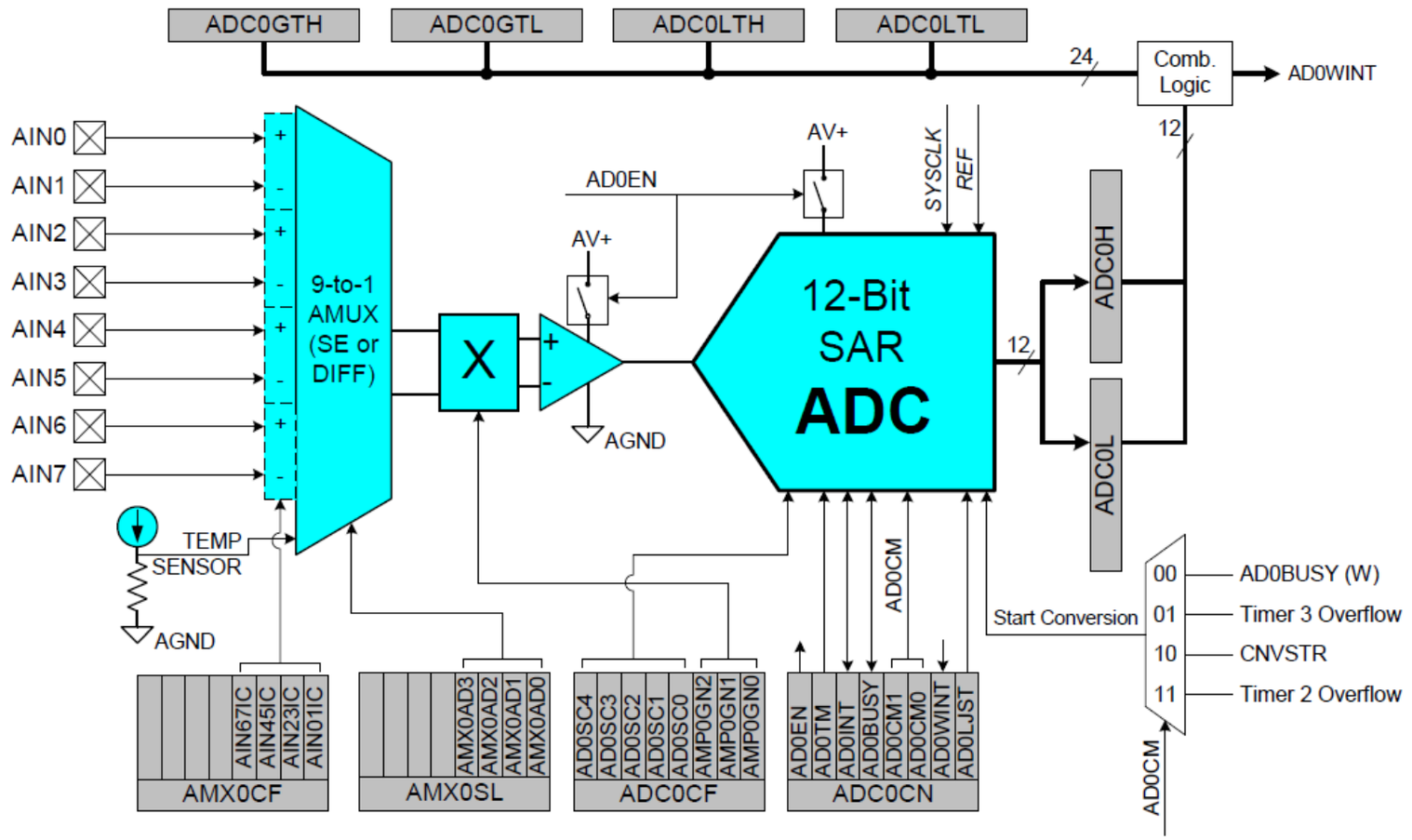
- If the value loaded in **ADC0CF** is 10000000, then the SAR0 conversion frequency will be 941 KHz approximately and the PGA0 gain will be set to 1

AMX0CF Configuration Register

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

AMX0SL Channel Selection Register

Bit	Symbol	Description
7-4	-	UNUSED. Read=0000, Write=don't care
3-0	**AMX0AD3-0**	***AMX0 Address Bits*** 0000-1111: ADC Inputs selected according to channel selection table on next slide.



AMUX0 Channel Selection- AMX0SL SFR

AMX0CF Bits 3-0	AMX0AD3-0								
	0000	0001	0010	0011	0100	0101	0110	0111	1xxx
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

ADC0 Control Register

- One of the main registers for activating and configuring the ADC0.
  - ADC0CN register (ADC Control Register), which contains settings bits and flag bits relevant to the ADC.
  - C8051F02x page 66 of C8051F02x's.pdf

Bit 7 AD0EN	Bit 6 AD0TM ADC Track Mode Bit	Bit 5 AD0INT Conversion Interrupt flag (must cleared by software)	Bit 4 AD0BUSY	Bit 3-2 AD0CM1-0	Bit 1 AD0WINT:	Bit 0 AD0LJST
AD0EN: ADC0 Enable Bit 1: ADC0 Enabled. ADC0 is active and ready for data conversion 0: ADC0 Disabled	1: Tracking Defined by ADSTM1-0 bits 0: Continuous tracking unless a conversion is in process	1: ADC0 has completed a data conversion 0: ADC0 has not completed a data conversion	1: ADC0 Conversion is in progress. 0: Conversion is complete or a conversion is not currently in progress. AD0INT is set to logic 1 on the falling edge of AD0BUSY.	Refer to the next slide	ADC0 Window Compare Interrupt Flag	0: Data in ADC0H:ADC0L registers are right-justified. 1: Data in ADC0H:ADC0L registers are left-justified.

ADC0 Start of Conversion Mode Select.

Bit 6
AD0TM
ADC Track Mode Bit
1: Tracking Defined by ADSTM1-0 bits
0: tracking is continuous unless a conversion is in process

Bit 3-2
AD0CM1-0

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.

Detecting ADC0 End-of-Conversion

- Polling Method
  - **AD0INT** bit (**ADC0CN.5**) may be polled to determine when a conversion has completed
  - Once the bit is set, read the ADC0 data

- Interrupt Method:
  - If ADC0 End-of-Conversion Interrupt (**EIE2.1**) and global interrupts are enabled, then an interrupt will be generated and the appropriate ADC0 ISR will be executed
  - Inside the ADC0 ISR, read the ADC0 data

C8051F020x ADC EXAMPLE

```
MOV REF0CN, #00000011B ;Enable internal bias, reference buffer,
                        ;VREF0 pin
MOV ADC0CF, #10000000B ;SAR0 Conversion clock=914 kHZ app.,
                        ; Gain 1

MOV AMX0CF, #00H       ;8 single-ended inputs
MOV AMX0SL, #00H       ;Selects AIN0
MOV ADC0CN, #10001101B ;Enable ADC0,
                       ; Continuous Tracking Mode, conversion
                       ; initiated on Timer 2 overflow
                       ; left justify
```

Final Exam REVIEW

- Exam time: 90 minutes, 14:20 – 16:00 PM, Wednesday, 18 April.

TEST DETAILS

- Will contain a mixture of the following question types:
  - Short answer
    - Answers no more than a few sentences.
  - Multiple choice
  - Numerical answer
  - Fill-in-the-blanks

The initial contents of Register A are 0b11110000
What are the contents of register A after the execution of an ANL instruction against A with an operand of 0b00110000?

0b11110000

0b00110000

0b00110000

Result is 1 only if both are 1

The initial contents of Register A are 0b10101010
What are the contents of register A after the execution of an ORL instruction against A with an operand of 0b01010101?

0b10101010

0b01010101

Result is 1 if either are 1

0b11111111

What is the difference between the RET and the RETI instruction's behaviour?

RET: pops PC value from the stack and decrements stack pointer by 2.
RETI: Same behaviour as RET, but first re-enables equal and lower-priority interrupts.

What two pin states can be used to trigger an external interrupt?

External interrupts can be triggered in response to the interrupt pin being at a low level or in response to a falling edge on the pin.

The A register has an initial value of 0b11110000
What are the register's contents after executing the following code?

```
RR  A
RR  A
RR  A
RR  A
```

0b00001111
(RR = rotate right)

Briefly describe the role, function, and use of a watchdog timer.

Role: prevents the system from hanging.
Function: resets the system if the timer is not reset in a timely manner.
Use: In a regularly-accessed region of code, the timer is reset in software. If this reset isn't reached (code is hanging), the system will reset

What addressing mode is used by the instruction `MOV A, #34H` ?

Immediate addressing mode.

What timer condition triggers a Timer-related interrupt?

The timer interrupt is triggered when a timer overflows.

Timer mode information and details about the TMOD register are provided. Write code (1 line) that sets Timer 0 to Mode 1.

	Mode name
MODE 0	13-bit timer mode: 8 bits of THx and 5 bits of TLx
MODE 1	16-bit timer mode. TLx counts 0-255; on overflow, this adds 1 to THx
MODE 2	8-bit timer mode. TLx auto-reloads with THx value.
MODE 3	“Split timer” mode: THx is one 8-bit timer, and TLx is another.

Table 31. TMOD Register

TMOD (S:89h)
Timer/Counter Mode Control Register

7	6	5	4	3	2	1	0
GATE1	C/T1#	M11	M01	GATE0	C/T0#	M10	M00

Timer 0 Mode Select Bit

M10	M00	Operating mode
0	0	Mode 0: 8-bit Timer/Counter (TH0) with 5-bit prescaler (TL0).
0	1	Mode 1: 16-bit Timer/Counter.
1	0	Mode 2: 8-bit auto-reload Timer/Counter (TL0) (2)
1	1	Mode 3: TL0 is an 8-bit Timer/Counter

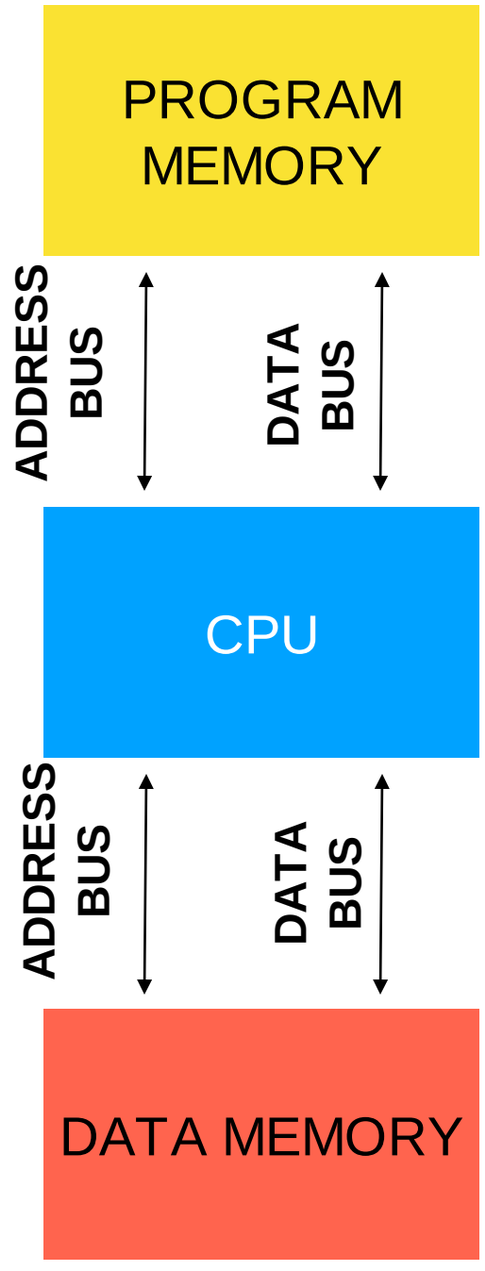
TH0 is an 8-bit Timer using Timer 1’s TR0 and TF0 bits.

```
MOV TMOD,#00000001B                ;TMOD reg: 0 0 0 0 0 0 0 1
```

Given the following .HEX file contents, circle the checksum column.

```
:10000000 7438   1136   1150   740E   1136   1150   7401   1136   B6
:10001000 1150   7406   1136   1150   7448   1143   1150   7445   33
:10002000 1143   1150   744C   1143   1150   744C   1143   1150   31
:10003000 744F   1143   80FE   F590   C2A0   C2A1   D2A2   1150   0C
:10004000 C2A222 F590   D2A0   C2A1   D2A2   1150   C2A222        75
:09005000 7B32   7CFF   DCFE   DBFA22                             AE
:00000001                                                        FF
```

What high-level computer architecture is illustrated in the following figure?



Harvard Architecture

A) In ‘standard’ 8051’s, how many clock cycles are there per machine cycle?
12

B) Given a 12 MHz Crystal, find the time in microseconds that one machine cycles takes.

12 MHz / 12 = 1 MHz
1/1 MHz = *1 μs*

When the PUSH instruction is executed, what register is incremented by 1?

The Stack Pointer (SP).