CS4101 Introduction to Embedded Systems

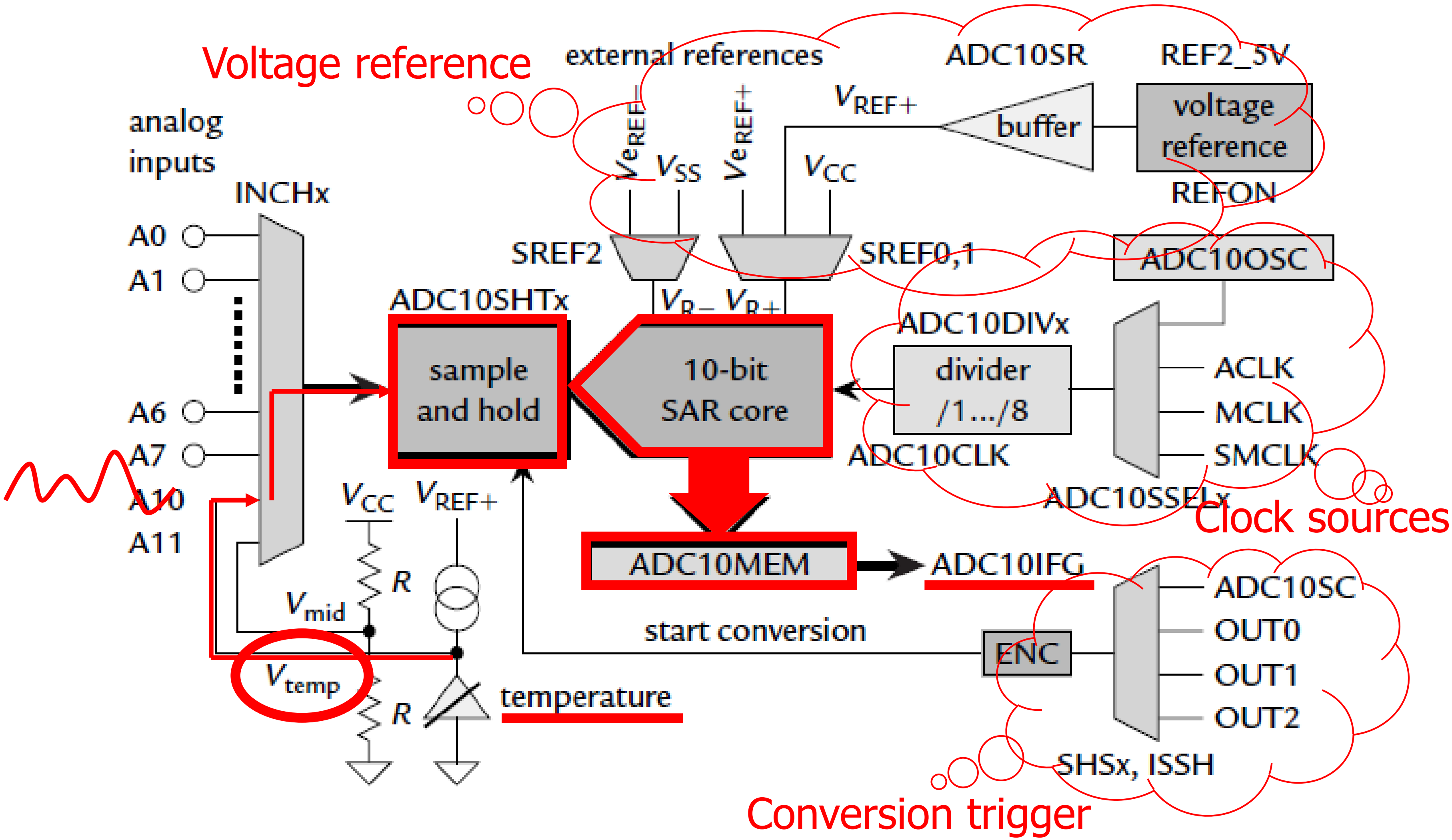
Lab 5: Analog-to-Digital Converter

Prof. Chung-Ta King
Department of Computer Science
National Tsing Hua University, Taiwan

Introduction

- In this lab, we will learn ADC of MSP430
  - Configuration of ADC10
  - Use of ADC10 to measure the temperature of LauchPad

Simplified Block Diagram of ADC10

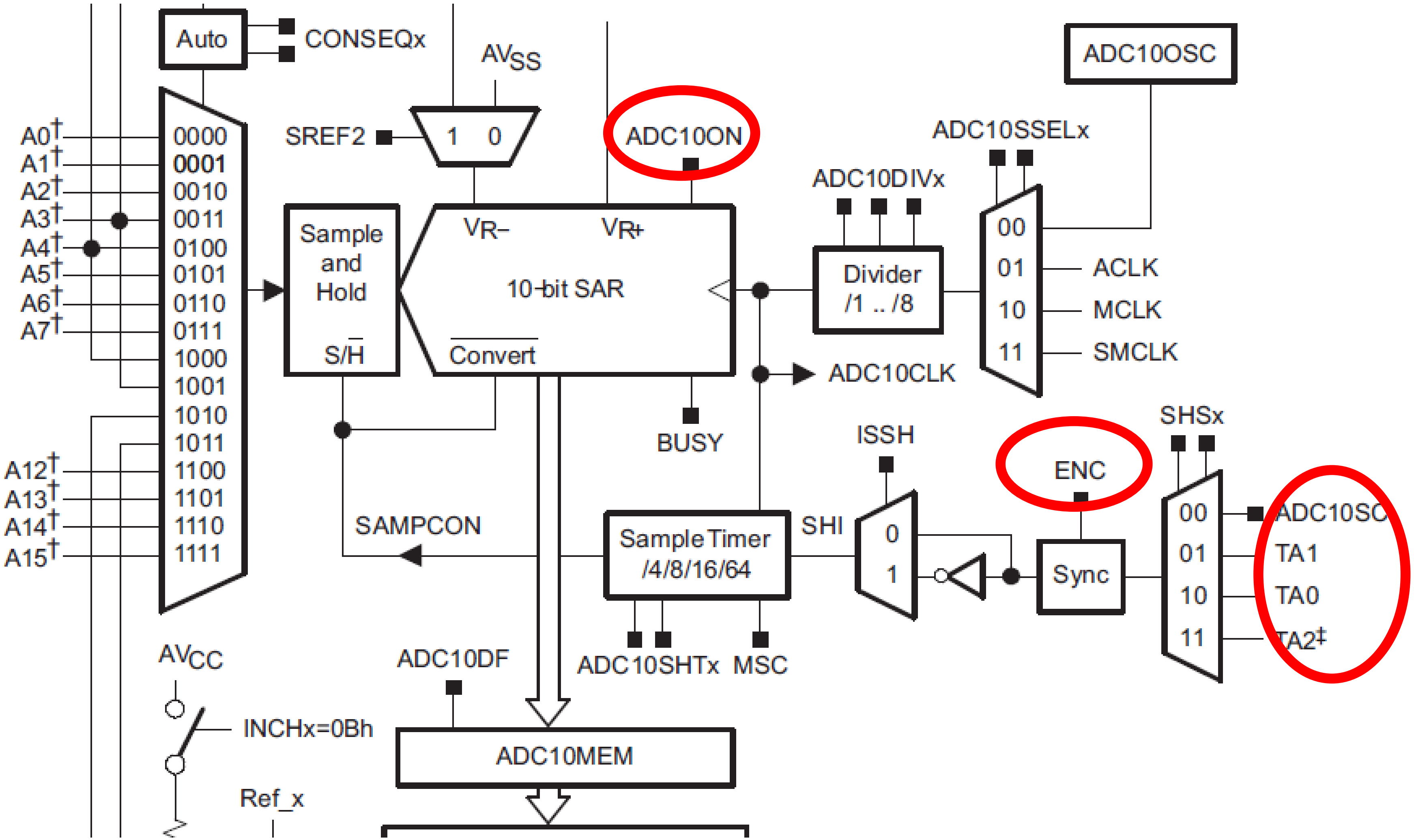


ADC10 Registers

Register	Short Form	Register Type	Addr.	Initial State
ADC10 input enable register 0	ADC10AE0	Read/write	04Ah	Reset with POR
ADC10 input enable register 1	ADC10AE1	Read/write	04Bh	Reset with POR
ADC10 control register 0	**ADC10CTL0**	Read/write	01B0h	Reset with POR
ADC10 control register 1	**ADC10CTL1**	Read/write	01B2h	Reset with POR
ADC10 memory	**ADC10MEM**	Read	01B4h	Unchanged
ADC10 data transfer control register 0		Read/write	048h	Reset with POR
ADC10 data transfer control register 1	ADC10DTC1	Read/write	049h	Reset with POR
ADC10 data transfer start address	ADC10SA	Read/write	01BCh	0200h with POR

Where the data is saved

Enabling Sampling and Conversion



Steps for Single Conversion

(1) Configure ADC10, including the ADC10ON bit to enable the module.

The ENC bit must be clear so that most bits in ADC10CTL0 and ADC10CTL1 can be changed.

(2) Set the ENC bit to enable a conversion.

This cannot be done if the module is being configured in (1).

(3) Trigger the conversion.

This is done either by setting the ADC10SC bit or by an edge from Timer_A.

- ADC10ON, ENC, ADC10SC are all in control register ADC10CTL0

ADC10CTL0

15	14	13	12	11	10	9	8
SREFx			ADC10SHTx		ADC10SR	REFOUT	REFBURST
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

7	6	5	4	3	2	1	0
MSC	REF2_5V	REFON	ADC10ON	ADC10IE	ADC10IFG	ENC	ADC10SC
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

Can be modified only when ENC = 0

SREFx Bits 15-13 Select reference

- 000 $V_{R+} = V_{CC}$ and $V_{R-} = V_{SS}$
- 001 $V_{R+} = V_{REF+}$ and $V_{R-} = V_{SS}$ — ideal for the temperature sensor
- 010 $V_{R+} = Ve_{REF+}$ and $V_{R-} = V_{SS}$
- 011 V_{R+} = Buffered Ve_{REF+} and $V_{R-} = V_{SS}$
- 100 $V_{R+} = V_{CC}$ and $V_{R-} = V_{REF-}/ V_{eREF-}$
- 101 $V_{R+} = V_{REF+}$ and $V_{R-} = V_{REF-}/ V_{eREF-}$
- 110 $V_{R+} = Ve_{REF+}$ and $V_{R-} = V_{REF-}/ V_{eREF-}$
- 111 V_{R+} = Buffered Ve_{REF+} and $V_{R-} = V_{REF-}/ V_{eREF-}$

ADC10SHTx Bits 12-11 ADC10 sample-and-hold time

- 00 4 × ADC10CLKs
- 01 8 × ADC10CLKs
- 10 16 × ADC10CLKs
- 11 64 × ADC10CLKs — ideal for the temperature sensor

ADC10CTL0 cont'd

REFON	Bit 5	Reference generator on 0 Reference off 1 Reference on
ADC10ON	Bit 4	ADC10 on 0 ADC10 off 1 ADC10 on
ADC10IE	Bit 3	ADC10 interrupt enable 0 Interrupt disabled 1 Interrupt enabled
ENC	Bit 1	Enable conversion 0 ADC10 disabled 1 ADC10 enabled
ADC10SC	Bit 0	Start conversion. Software-controlled sample-and-conversion start. ADC10SC and ENC may be set together with one instruction. ADC10SC is reset automatically. 0 No sample-and-conversion start 1 Start sample-and-conversion

```
ADC10CTL0 = SREF_2 + ADC10SHT_1;  // Reference range & SH time
```

ADC10CTL1

15	14	13	12	11	10	9	8
INCHx				SHSx		ADC10DF	ISSH
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

7	6	5	4	3	2	1	0
ADC10DIVx			ADC10SSELx		CONSEQx		ADC10BUSY
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	r-0

(shaded) Can be modified only when ENC = 0

INCHx Bits 15-12 Input channel select. These bits select the channel for a single-conversion or the highest channel for a sequence of conversions.

0000	A0
0001	A1
0010	A2
0011	A3
0100	A4
0101	A5
0110	A6
0111	A7
1000	V_{eREF+}
1001	V_{REF-}/V_{eREF-}
1010	Temperature sensor
1011	$(V_{CC} - V_{SS}) / 2$
1100	$(V_{CC} - V_{SS}) / 2$, A12 on MSP430x22xx devices
1101	$(V_{CC} - V_{SS}) / 2$, A13 on MSP430x22xx devices
1110	$(V_{CC} - V_{SS}) / 2$, A14 on MSP430x22xx devices
1111	$(V_{CC} - V_{SS}) / 2$, A15 on MSP430x22xx devices

ADC10CTL1 cont'd

SHSx	Bits 11-10	Sample-and-hold source select	
		00	ADC10SC bit
		01	Timer_A.OUT1
		10	Timer_A.OUT0
		11	Timer_A.OUT2 (Timer_A.OUT1 on MSP430x20x2 devices)
ADC10DF	Bit 9	ADC10 data format	
		0	Straight binary
		1	2s complement
ISSH	Bit 8	Invert signal sample-and-hold	
		0	The sample-input signal is not inverted.
		1	The sample-input signal is inverted.
ADC10DIVx	Bits 7-5	ADC10 clock divider	
		000	/1
		001	/2
		010	/3
		011	/4
		100	/5
		101	/6
		110	/7
		111	/8
ADC10SSELx	Bits 4-3	ADC10 clock source select	
		00	ADC10OSC
		01	ACLK
		10	MCLK
		11	SMCLK

CONSEQx	Bits 2-1	Conversion sequence mode select	
		00	Single-channel-single-conversion
		01	Sequence-of-channels
		10	Repeat-single-channel
		11	Repeat-sequence-of-channels

```
ADC10CTL1 = INCH_10 + ADC10DIV_0; // Temp Sensor ADC10CLK
```

Sample Code 1 for ADC10

- Repetitive single conversion:
  - A single sample is made on A1 with reference to Vcc
  - If A1 > 0.5*Vcc, P1.0 set, else reset.
  - Software sets ADC10SC to start sample and conversion. ADC10SC automatically cleared at end of conversion.
  - Use ADC10 internal oscillator to time the sample and conversion.

Sample Code 1 for ADC10

```
#include "msp430.h"
void main(void) {
  WDTCTL = WDTPW + WDTHOLD;    // Stop WDT
  // H&S time 16x, interrupt enabled
  ADC10CTL0 = ADC10SHT_2 + ADC10ON + ADC10IE;
  ADC10CTL1 = INCH_1;     // Input from A1
  ADC10AE0 |= 0x02; // Enable pin A1 for analog in
  P1DIR |= 0x01;    // Set P1.0 to output
  ADC10CTL0 |= ENC + ADC10SC; // Start sampling
  for (;;)  { }
}
#pragma vector=ADC10_VECTOR
__interrupt void ADC10_ISR(void) {
  if (ADC10MEM < 0x1FF) P1OUT &= ~0x01;
  else P1OUT |= 0x01;
  ADC10CTL0 |= ENC + ADC10SC; // enable sampling
}
```

Sample Code 2 for ADC10

- Continuous sampling driven by Timer_A
  - A1 is sampled 16/second (ACLK/2048) with reference to 1.5V, where ACLK runs at 32 KHz driven by an external crystal.
  - If A1 > 0.5Vcc, P1.0 is set, else reset.
  - Timer_A is run in up mode and its CCR1 is used to automatically trigger ADC10 conversion, while CCR0 defines the sampling period
  - Use internal oscillator times sample (16x) and conversion (13x).

Sample Code 2 for ADC10

```
#include  "msp430.h"
int i=1;
void main(void) {
  WDTCTL = WDTPW + WDTHOLD; // Stop WDT
  // TA1 trigger sample start
  ADC10CTL1 = SHS_1 + CONSEQ_2 + INCH_1;
  ADC10CTL0 = SREF_1 + ADC10SHT_2 + REFON +
              ADC10ON + ADC10IE;
  __enable_interrupt();  // Enable interrupts
  TACCR0 = 30;        // Delay for Volt Ref to settle
  TACCTL0 |= CCIE; // Compare-mode interrupt
  TACTL = TASSEL_2 + MC_1; // SMCLK, Up mode
  while(i);                // Wait for settle
  TACCTL0 &= ~CCIE;       // Disable timer Interrupt
  __disable_interrupt();
```

Sample Code 2 for ADC10

```
  ADC10CTL0 |= ENC;       // ADC10 Enable
  ADC10AE0 |= 0x02;       // P1.1 ADC10 option select
  P1DIR |= 0x01;          // Set P1.0 output
  TACCR0 = 2048-1;        // Sampling period
  TACCTL1 = OUTMOD_3;     // TACCR1 set/reset
  TACCR1 = 2046;          // TACCR1 OUT1 on time
  TACTL = TASSEL_1 + MC_1;       // ACLK, up mode

  while(1);
}
```

Timer_A CCR1 out mode 3: The output (OUT1) is set when the timer *counts* to the TACCR1 value. It is reset when the timer *counts* to the TACCR0 value.

Sample Code 2 for ADC10

```
  // ADC10 interrupt service routine
#pragma vector=ADC10_VECTOR
__interrupt void ADC10_ISR(void){
  if (ADC10MEM < 0x155) // ADC10MEM = A1 > 0.5V?
    P1OUT &= ~0x01;     // Clear P1.0 LED off
  else
    P1OUT |= 0x01;      // Set P1.0 LED on
}

#pragma vector=TIMERA0_VECTOR
__interrupt void ta0_isr(void){
  TACTL = 0;
  i = 0
}
```

Basic Lab

- Measure the temperature of MSP430 every second using the temperature sensor inside ADC10. Flash both LEDs if the temperature remains unchanged between two consecutive measurements. Flash the red LED if the temperature rises and the green LED if it drops.
  - The sampling of ADC10 must be triggered continuously by Timer_A.
  - You can use an infinite loop to flash the LEDs.

Suggested Operations

① Disable watchdog timer

② Set DCO as the source of SMCLK, 1 MHz, and VLO as ACLK

③ Set up both LED lights and set them initially off

④ You can modify the sample code 2 to complete the basic lab. You can first set timer_A source from SMCLK (DCO) to delay for 30 microsecond until the reference voltage is stable, and then set timer_A source from ACLK (VLO).

Suggested Operations

⑤ Configure ADC10 as follows:
   ① Set sample-and-hold source from Timer_A
   ② Use temperature sensor channel
   ③ Use ideal reference
   ④ Set conversion sequence mode as repeat-single-channel
   ⑤ Enable ADC10 interrupt

⑥ ADC10IFG is set every second when conversion results(temperature) are loaded into ADC10MEM, which in turn triggers ADC10 ISR.

Bonus

- Based on Basic 1, if the temperature is higher than 737, measure the temperature and flash the red LED at 5 Hz. When the temperature is below 737, then return to Basic 1.