Programming for Embedded Systems
Lecture 7: ADC and the Analog Comparator

Bernhard Firner

March 2, 2015

Analog Input

- The G2553 has two kinds of analog input
- A 10 bit analog-to-digital convert (ADC10)
  - Converts an analog input in a given range to a fractional value
- A voltage level comparator (Comparator A+)
  - Triggers an interrupt when a specified voltage level is detected

Uses of Analog to Digital Conversion

- Any kind of digital recording
  - Audio and video
- Any time analog components are used as sensors
  - Temperature, humidity, etc

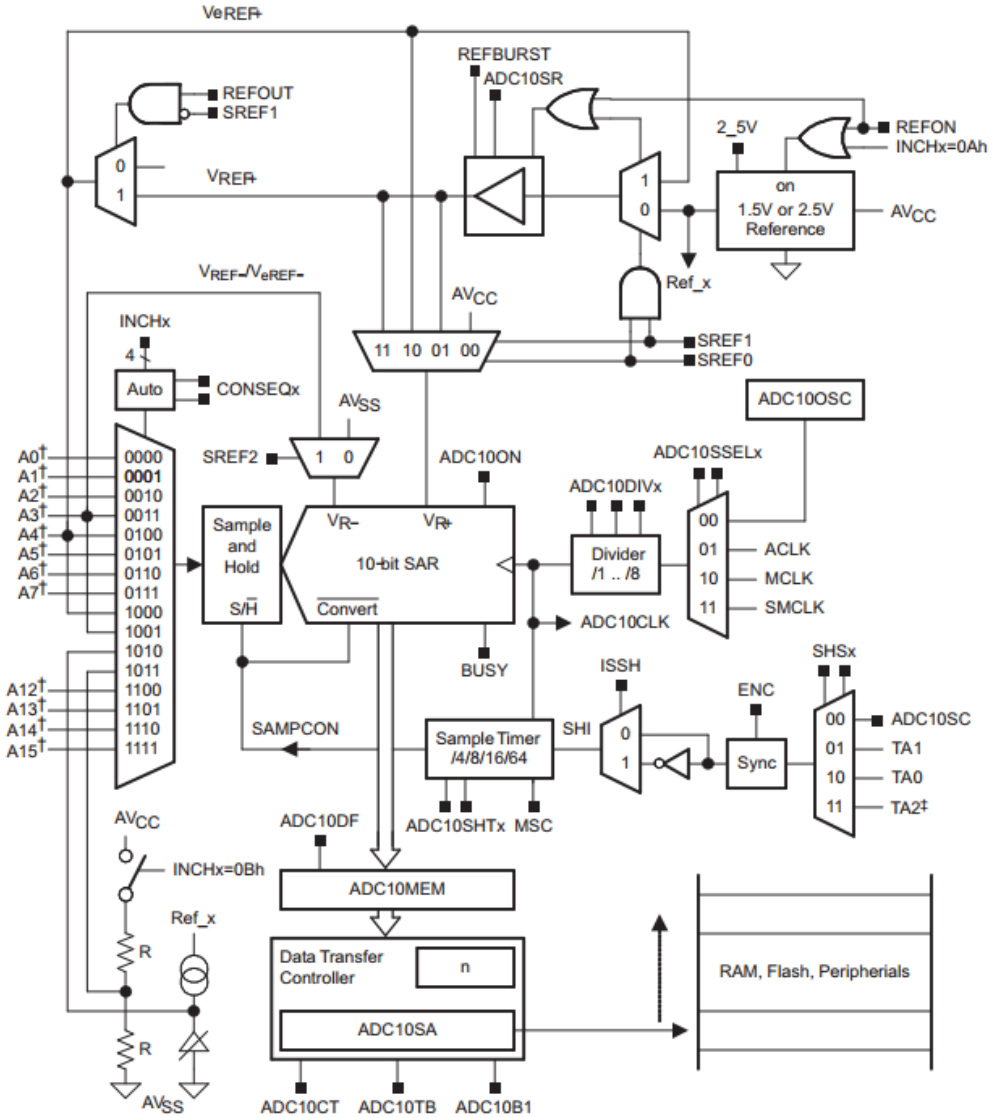
ADC10

- ADC10 gives a value that is based upon a reference

$$value = \frac{input}{reference}$$

- ADC10 has 10 bits of precision
  - R^+ and R^- are the upper and lower limits (0 and 1024)
  - V_{in} is measured like this:
    - $value = 1023 \frac{V_{in} - V_{R-}}{V_{R+} - V_{R-}}$
  - There are 1024 quantizations from 0 to 1
  - If V_{in} is beyond the limits the value clips to 0 or 1023

ADC10 Block Diagram (from TI SLAU144J)



†Channels A12-A15 are available in MSP430F22xx devices only. Channels A12-A15 tied to channel A11 in other devices. Not all channels are available in all devices.
‡TA1 on MSP430F20x2, MSP430G2x31, and MSP430G2x30 devices

Settings

- ADC10 has a lot of settings
- Most of them are in registers ADC10CTL0 and ADC10CTL1
- ADC10AE0-1 are used to set which pins are analog input
  - For example, this sets Pin 1.5 as analog input:

    ADC10AE0 |= BIT5;

Settings a Voltage Reference

- You can set the sources for V_{R-} and V_{R+}
  - 1.5V or 2.5V internal references
  - External source on an input pin
  - V_{cc} or V_{ss}
- Bits 15-13 in ADC10CTL0 set the reference
- Bits 6-5 control the 1.5V and 2.5V reference

Setting a Timer Source

- You can source the converter with different clocks
- Bits 4-3 in ADC10CTL1 control the clock source
- Bits 7-5 set the divider

The Input Source

- The input source is selected in bits 15-12 of ADC10CTL1.
- There are 8 possible external pins
- There are also several interval comparisons and sources

Conversion Modes

- We're not done yet! How do we want to use the converter?
  - Single conversion of one input?
  - Multiple conversions on several inputs?
  - Repeatedly convert a single input?
  - Repeatedly convert multiple inputs?
- All options controlled by bits 2-1 of ADC10CTL1
- The inputs used in multiple conversion modes are controlled by other registers

The Results

- The results of a conversion are placed in ADC10MEM
- Bit 4 of ADC10CTL0 turns on the ADC
- Bit 3 enables an interrut for when the conversion completes
- Bit 1 enables conversions
- Writing 1 to Bit 0 starts the conversion

Conversion Example

```
void main(void) {
    //Stop the watchdog timer
    WDTCTL = WDTPW + WDTHOLD;

    //Set up Pin 1.5 as analog input
    P1DIR &= ~BIT5;
    P1SEL |= BIT5;
    ADC10AE0 |= BIT5;

    //Set up ADC10; source from VCC and VSS,
    //turn on the ADC core, and enable the interrupt
    ADC10CTL0 = SREF_0 | ADC10ON | ADC10IE;
    //Input on channel 5 (PIN 1.5). Divide the clock by 8.
    ADC10CTL1 = INCH_5 + ADC10DIV_3;

    //Ready the interrupts
    __enable_interrupt();
```

Starting the Conversion

```
    //Enable sampling/conversion and turn on encoding to begin
    ADC10CTL0 |= ENC | ADC10SC;
    //Stop the CPU and wait for conversion to finish (LPM0)
    LMP0;
    //Read the result
    volatile int result = ADC10MEM;
    //You can set a breakpoint here to
    //read the variable through the debugger
    //Making the variable volatile makes sure its value is preserved
    while (1);
}
```

The Interrupt

```
// ADC10 interrupt service routine so we sleep
//while waiting for the conversion to finish
#pragma vector=ADC10_VECTOR
__interrupt void ADC10_ISR(void) {
    //Turn the CPU back on
    LPM0_EXIT;
}
```

- After the interrupt ADC10MEM holds the result

Voltage Comparator

- The MSP430 also has a voltage comparator
- Can be used to trigger interrupts when an input goes above or below a reference
- We won't be using it, so for brevity we can skip its details
- The next lab will use the ADC10 module