LINKÖPING UNIVERSITY

TNE097 Micro Computer Systems

LAB 3 Analog-to-Digital Converter

Goals

- Understand basic principle and knowledge of ADC
- Able to do basic configuration on MSP430 ADC
- Use DTC to store and manage ADC reading

Task preparation

This lab contains four tasks. Prepare the code of Task 1 – Task 4 and submit your lab preparation via Lisam before the lab. Codes of the tasks will be checked by lab assistants during the lab session.

Task 1

The following code shows that MSP430 uses internal 1.5V reference voltage to convert voltage from channel A1.

```
//==============================================================

#include "msp430g2553.h"

void main(void)

{

  WDTCTL = WDTPW + WDTHOLD;                 // Stop WDT

  ADC10CTL0 = SREF_1 + ADC10SHT_2 + REFON + ADC10ON;

  ADC10CTL1 = INCH_1;                       // input A1, pin P1.1

  ADC10AE0 |= 0x02;                         // Select ADC function for
pin P1.1

  P1DIR |= 0x01;                            // Set P1.0 to output
direction

  for (;;)

  {

   ADC10CTL0 |= ENC + ADC10SC;        // Sampling and conversion start

   while(ADC10CTL1 & ADC10BUSY) {}          //wait until the
conversion is ready

                                 // Testing ADC10BUSY bit in ADC10CTL1

    if (ADC10MEM < 0x88)                    // ADC10MEM = A1 >   V?

      P1OUT &= ~0x01;                       // Clear P1.0 LED off

    else

      P1OUT |= 0x01;                        // Set P1.0 LED on

  }

}

//==============================================================
```

Task 1.1 Modify the code so that MSP430 measures analog voltage from channel A5 and change reference voltage to internal reference 2.5V.

Task 1.2 Modify the code so that MSP430 uses interrupt to read the measured voltage. If the measured voltage is higher than 1.5V turn on RED LED by setting P1.0 (to 1). Otherwise, reset P1.0 (to 0).

Task 2

The following code use Timer0_A TACCR0 and TACCR1 to generate a pulse (PWM). The pulse can be utilized as the signal to start sampling and conversion for ADC10 of MSP430.

```
//======================================================================

       TACCR0 = 2048-1;                          // PWM Period

       TACCTL1 = OUTMOD_3;                       // Output Mode 3:
     set/reset

       TACCR1 = 1024-1;                          // TACCR1 PWM Duty
     Cycle

       TACTL = TASSEL_2 + MC_1;                  // SMCLK, up mode

//======================================================================
```

Task 2.1

Change the code from **Task 1** such that

- The ADC sampling signal is controlled by the Timer0_A CCR1 output unit instead of ADC10SC.
- Configure the ADC working mode to “Repeat single channel mode” from A1 so that ADC10 can be continuously triggered by Timer0_A output.
- Read the ADC10 value from ADC interrupts.
- When input voltage from A1 is higher than 1.5V, set P1.0, otherwise, reset P1.0.

Task 3

Implement the function that MSP430 use DTC mode to get multiple ADC readings from one channel. (The state machine of DTC one block mode is shown in the Figure 22-10.)

- Configure ADC10 as “repeat single channel mode” via A1, using internal reference 2.5V.
- Enable MSC bit to set ADC work continuously.
- Configure the ADC DTC mode as “one block” mode, 16 readings (n = 0x10) and assigned the read value to an array with 16 elements (as shown in Lecture 4, page 34).
- Handle the ADC DTC reading in the interrupt and calculate the average value of the ADC reading in the defined array.
- Restart the DTC in the ADC interrupt.

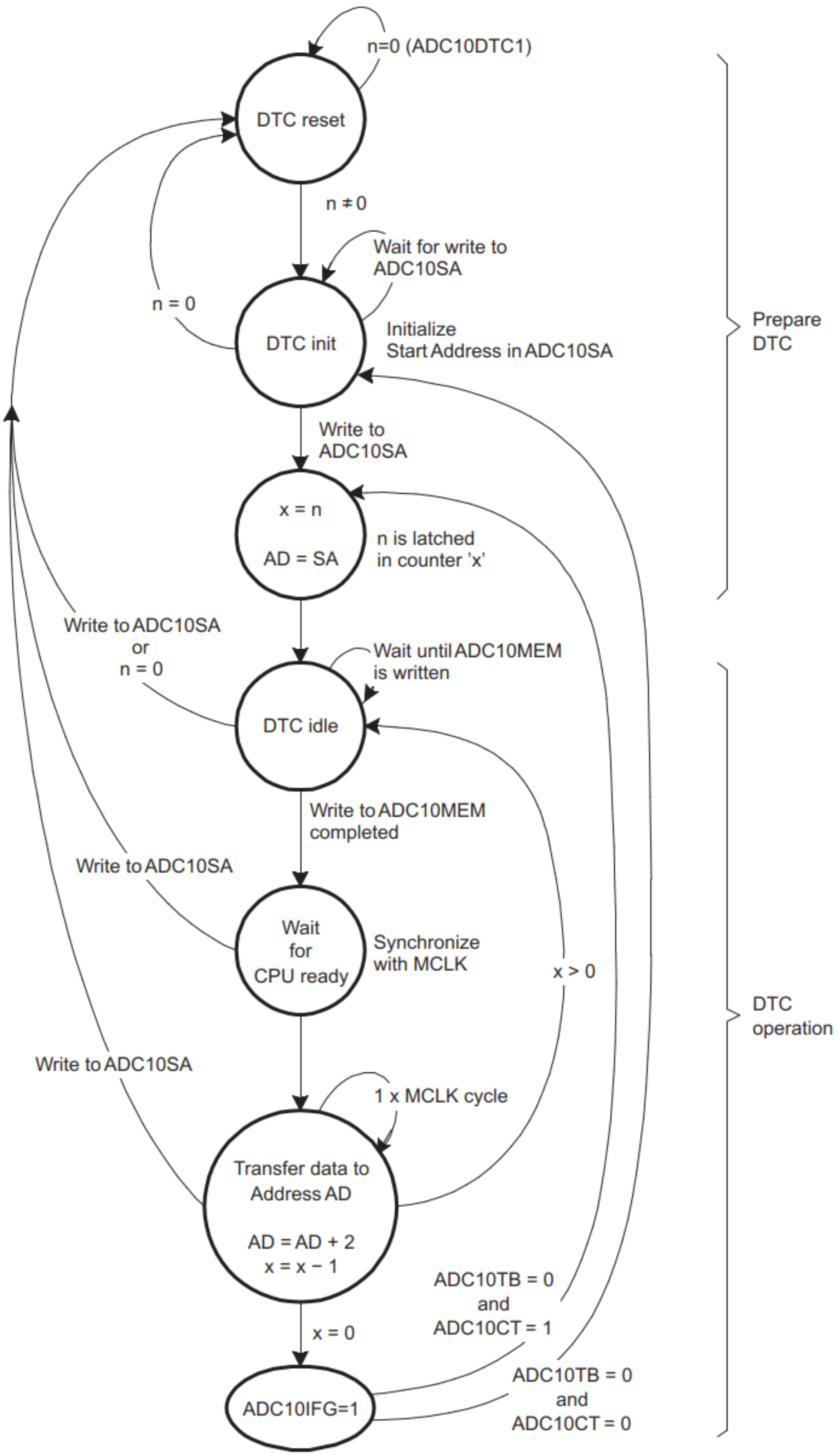
n=0 (ADC10DTC1)
DTC reset
n ≠ 0
Wait for write to ADC10SA
n = 0
DTC init
Initialize Start Address in ADC10SA
Prepare DTC
Write to ADC10SA
x = n
AD = SA
n is latched in counter 'x'
Write to ADC10SA or n = 0
Wait until ADC10MEM is written
DTC idle
Write to ADC10MEM completed
Write to ADC10SA
Wait for CPU ready
Synchronize with MCLK
x > 0
DTC operation
Write to ADC10SA
1 x MCLK cycle
Transfer data to Address AD
AD = AD + 2
x = x − 1
ADC10TB = 0 and ADC10CT = 1
x = 0
ADC10TB = 0 and ADC10CT = 0
ADC10IFG=1

Figure 22-10. State Diagram for Data Transfer Control in One-Block Transfer Mode