11012 ADV

Advanced 16-bit Peripheral Configuration

© 2007 Microchip Technology Incorporated. All Rights Reserved.

UNIVERSITY OF MICROCHIP UOFM MASTERS 2007

Class Objective

When you finish this class you will:

- Know how to configure and use ADC in your application
- Be able to use the ECAN™ module for communications
- Become comfortable in using DMA for transferring data to/from the ADC and the ECAN module

© 2007 Microchip Technology Incorporated. All Rights Reserved.

UNIVERSITY OF MICROCHIP UOFM MASTERS 2007

Agenda

I. DMA

- DMA Overview
- Configuration & Setup

II. 10/12-bit ADC

- ADC Overview
- Configuration & Setup
  - **ADC Inputs to Sample/Hold Association**
  - **Sample/Hold Sequence Configuration**
  - **ADC Triggers Configuration**
  - **Output Configuration**
- Application Examples & Exercises

© 2007 Microchip Technology Incorporated. All Rights Reserved.

University of Microchip UofM MASTERs 2007

Agenda

III. ECAN™ Module

- ECAN Overview
  - **Typical ECAN System**
  - **Message Buffer Structure**
  - **Interaction with DMA**
  - **Tx Process**
  - **RX Process**
- ECAN Lab

© 2007 Microchip Technology Incorporated. All Rights Reserved.

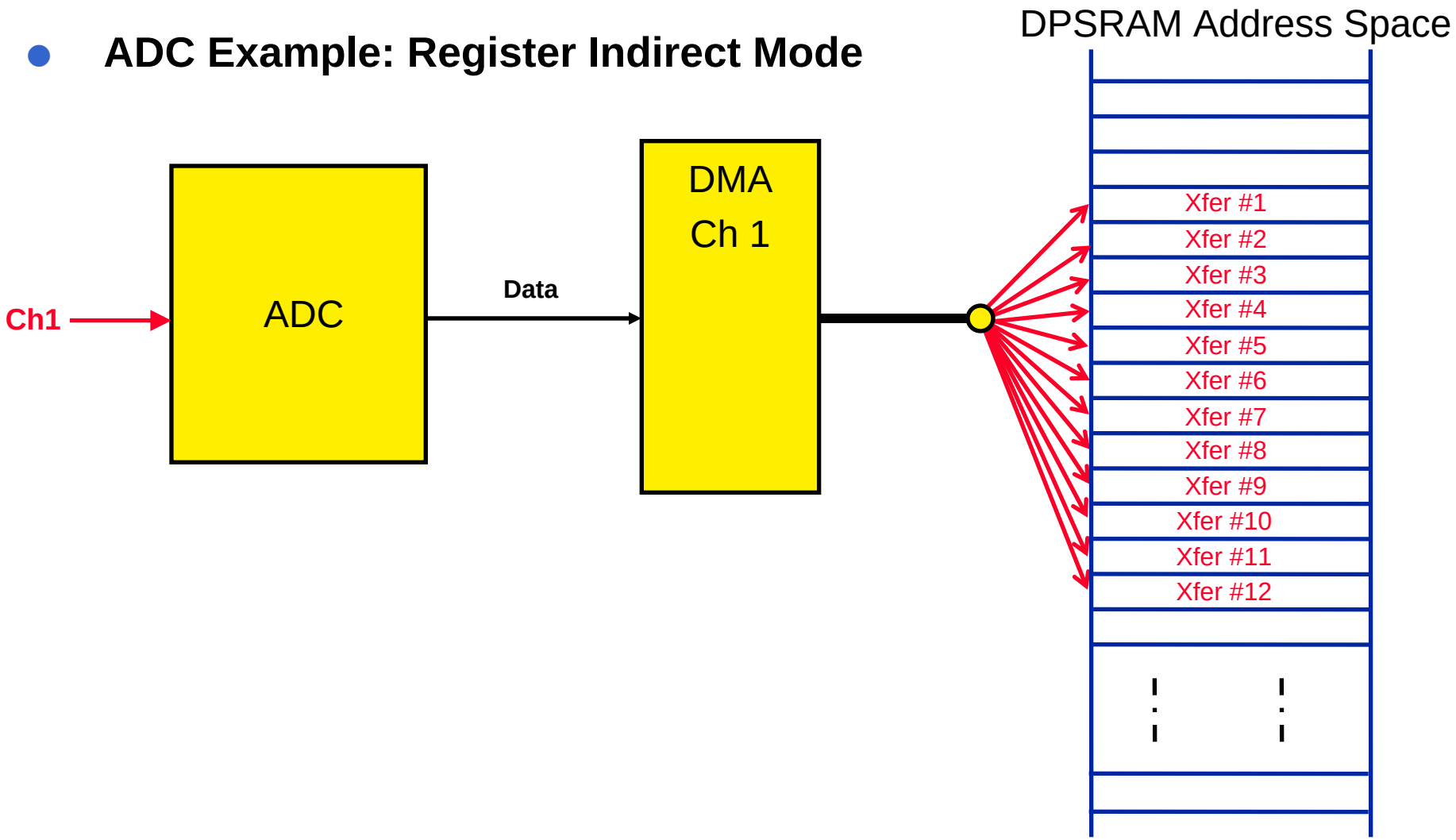
DMA

Direct Memory Access

© 2007 Microchip Technology Incorporated. All Rights Reserved.

DMA Example: ADC

- **ADC Example: Register Indirect Mode**



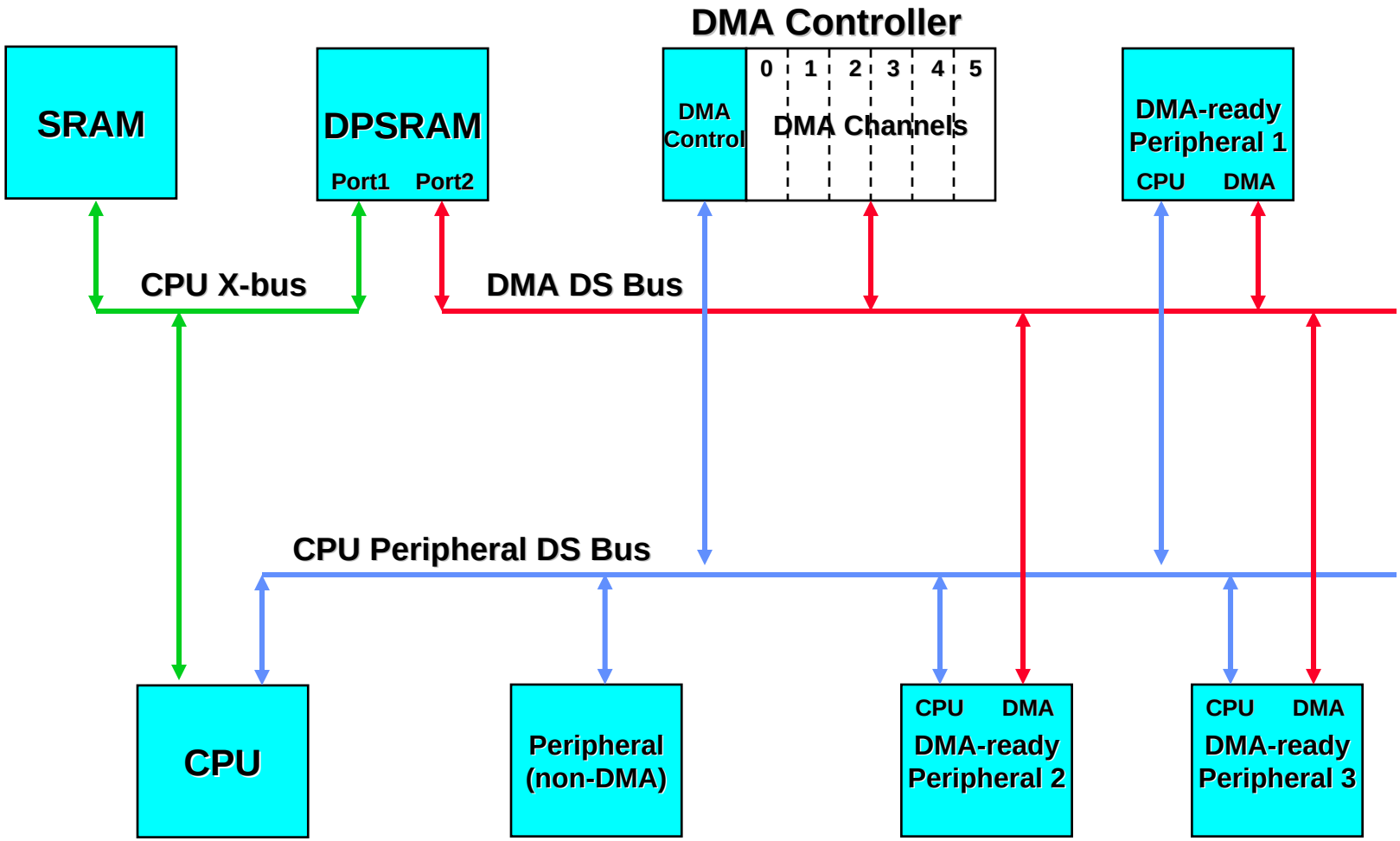
© 2007 Microchip Technology Incorporated. All Rights Reserved.

DMA Features

- **8 DMA channels**
- **Register indirect with post increment addressing mode**
- **Peripheral indirect addressing mode**
  - Peripheral generates destination address
- **CPU interrupt after half or full block transfer complete**
- **Byte or word transfers**
- **Fixed priority channel arbitration**
- **Manual or Automatic transfers**
- **One-shot or Auto-Repeat transfers**
- **'Ping-pong' mode**
  - Automatic switch between two buffers
- **DMA request for each channel can be selected from any supported interrupt sources**
- **Debug support features**

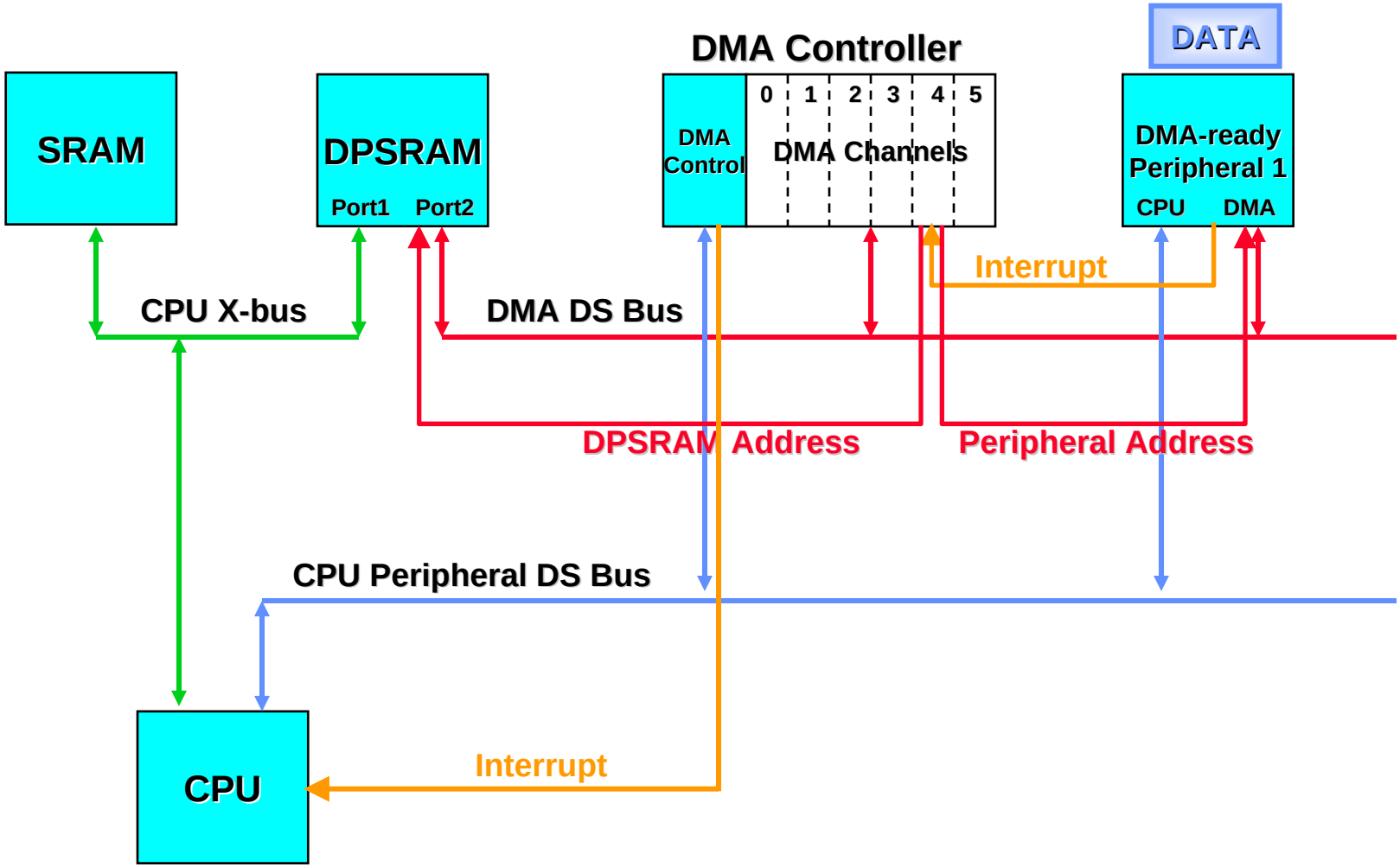
© 2007 Microchip Technology Incorporated. All Rights Reserved.

DMA Controller Block Diagram



© 2007 Microchip Technology Incorporated. All Rights Reserved.

DMA Controller Operation



© 2007 Microchip Technology Incorporated. All Rights Reserved.

University of Microchip UofM MASTERs 2007

DMA Support

- **Supported 16-bit Families**
  - PIC24H
  - dsPIC33F
- **Supported 16-bit Peripherals**
  - ECAN™ Module
  - Data Converter Interface (DCI)
  - 10-bit/12-bit A/D Converter
  - Serial Peripheral Interface (SPI)
  - UART
  - Input Capture
  - Output Compare
- **DMA Request Support Only**
  - Timers
  - External Interrupts

© 2007 Microchip Technology Incorporated. All Rights Reserved.

UNIVERSITY OF MICROCHIP UOFM MASTERS 2007

Enabling DMA Operation

1. Associate DMA channel with peripheral
2. Configure DMA capable peripheral
3. Initialize DPSRAM data start addresses
4. Initialize DMA transfer count
5. Select appropriate addressing and operating modes

© 2007 Microchip Technology Incorporated. All Rights Reserved.

Step 1: Associate DMA and Peripheral

- Associate peripheral IRQ with DMA via DMAxREQ
- Provide peripheral read/write address via DMAxPAD

Example: Associate DMA Channel 0 and 1 with UART2 Transmitter and Receiver respectively

```
DMA0REQbits.IRQSEL = 0x1F;
DMA0PAD = (volatile unsigned int) &U2TXREG;

DMA0REQbits.IRQSEL = 0x1E;
DMA0PAD = (volatile unsigned int) &U2RXREG;
```

© 2007 Microchip Technology Incorporated. All Rights Reserved.

Step 2: Configure DMA-Ready Peripheral

- Configure peripherals to generate interrupt for every transfer (if applicable)

Example: Configure UART2 to generate DMA request after each Tx and Rx character

```
U2STAbits.UTXISEL0 = 0;   // Interrupt after one Tx character is transmitted
U2STAbits.UTXISEL1 = 0;
U2STAbits.URXISEL  = 0;   // Interrupt after one RX character is received
```

- Enable Error interrupts (if applicable)

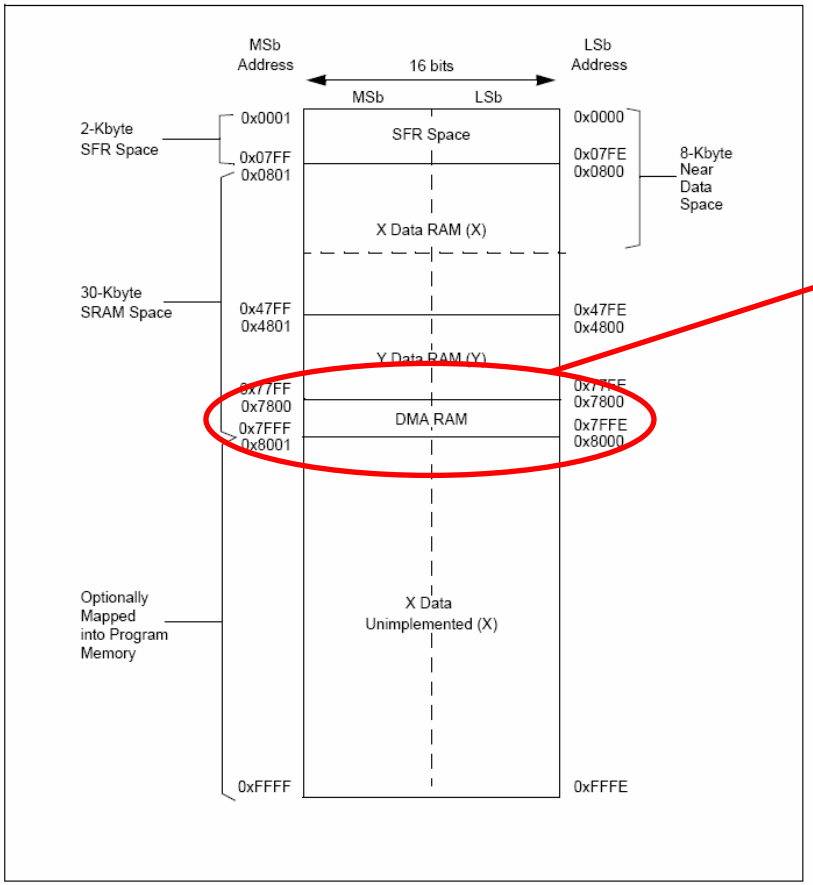
Example: Enable and process UART2 error interrupts

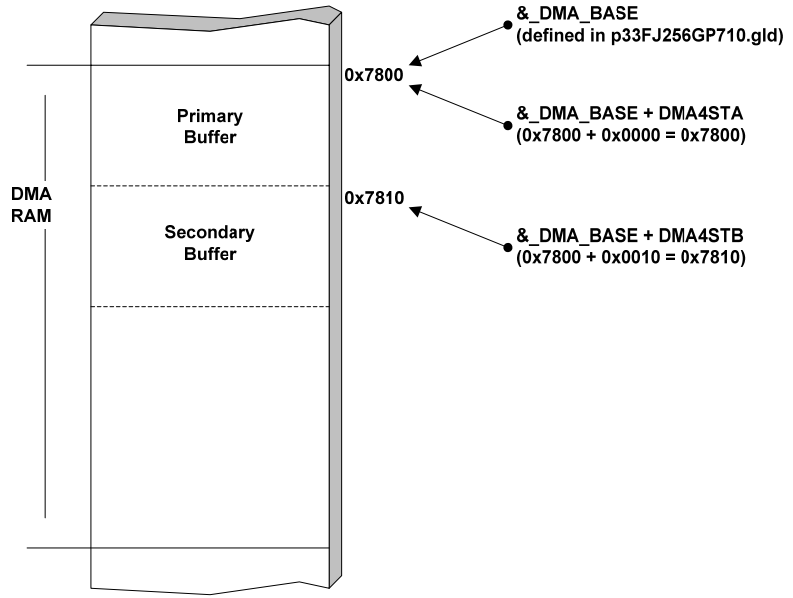
```
IEC4bits.U2EIE = 0;                  // Enable UART2 Error Interrupt

void __attribute__((__interrupt__)) _U2ErrInterrupt(void)
{
      /* Process UART 2 Error Condition here */

      IFS4bits.U2EIF = 0; // Clear the UART2 Error Interrupt Flag
}
```

© 2007 Microchip Technology Incorporated. All Rights Reserved.

Step 3: Initialize DPSRAM data start addresses

FIGURE 3-5: DATA MEMORY MAP FOR dsPIC33F DEVICES WITH 30 KBs RAM



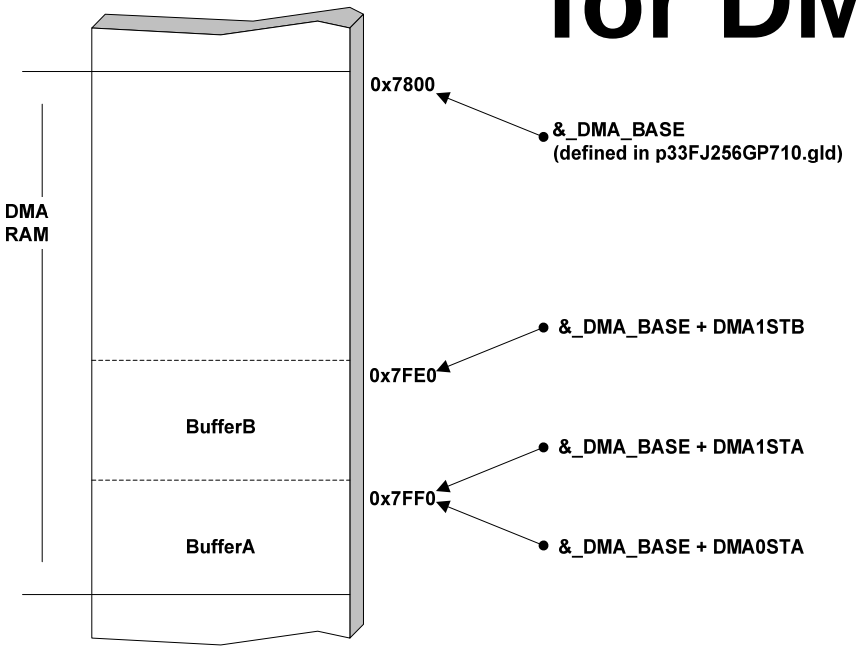


Example: Setup Primary and Secondary DMA Channel 4 buffers at 0x7800 and 0x7810

```
DMA4STA = 0x0000;
DMA4STB = 0x0010;
```

© 2007 Microchip Technology Incorporated. All Rights Reserved.

Step 3: MPLAB® IDE Support for DMA



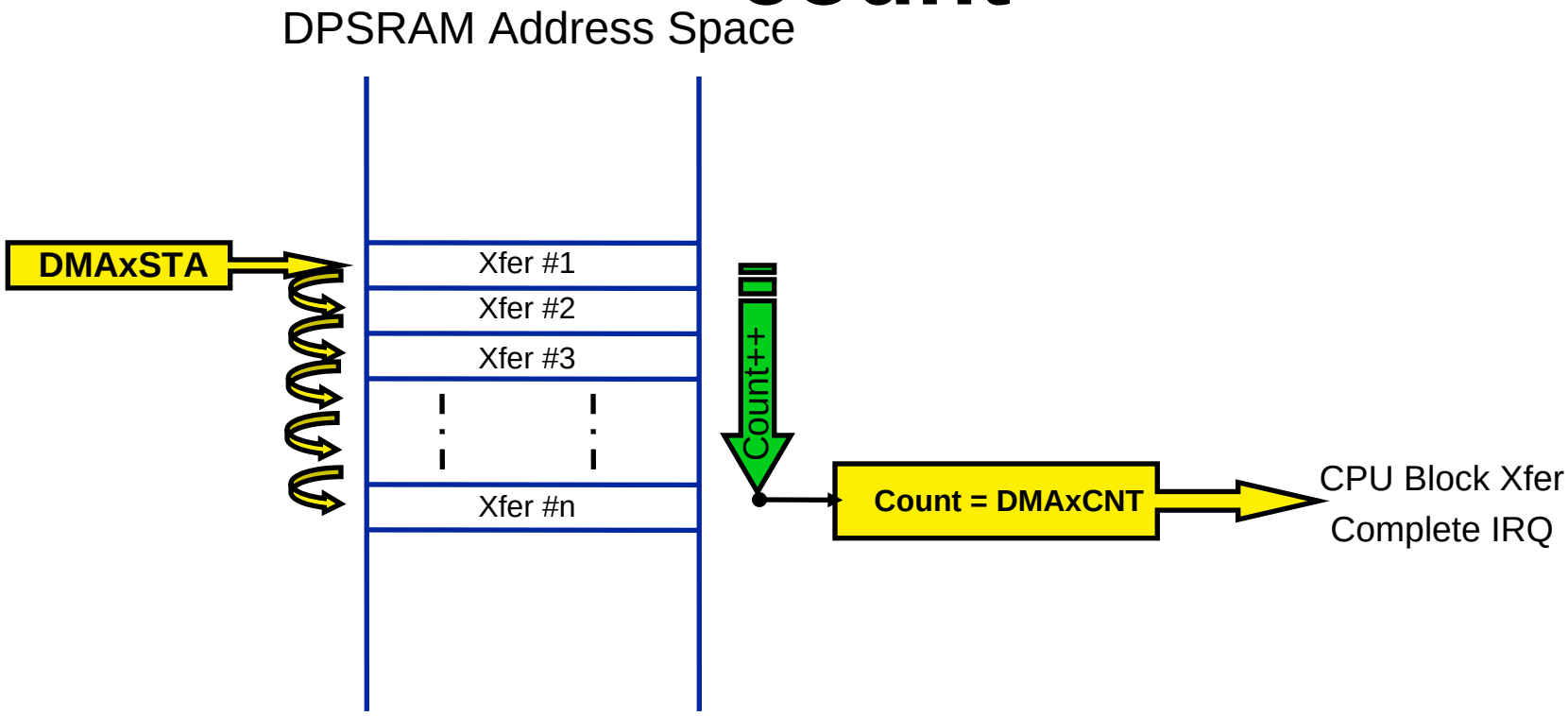
- Use __attribute__(space(dma)) and __builtin_dmaoffset()

Example: Allocate two buffers 8 words each in DMA memory for DMA Channel 1; Associate DMA Channel 0 with one of the buffers as well

```
unsigned int BufferA[8] __attribute__(space(dma));
unsigned int BufferB[8] __attribute__(space(dma));

DMA1STA = __builtin_dmaoffset(BufferA);
DMA1STB = __builtin_dmaoffset(BufferB);

DMA0STA = __builtin_dmaoffset(BufferA);
```

© 2007 Microchip Technology Incorporated. All Rights Reserved.

Step 4: Initialize DMA transfer count



Example: Setup DMA Channel 0 and 1 to handle 8 DMA requests

```
DMA0CNT = 7;   // 8 DMA Requests
DMA1CNT = 7;   // 8 DMA Requests
```

© 2007 Microchip Technology Incorporated. All Rights Reserved.

Step 5: Select appropriate DMA addressing and operating modes

- Word or byte size data transfers
- Peripheral to DPSRAM, or DPSRAM to peripheral transfers
- Post-increment or static DPSRAM addressing
- One-shot or continuous block transfers
- Interrupt the CPU when the transfer is half or fully complete
- Auto switch between two start addresses offsets (DMAxSTA or DMAxSTB) after each transfer complete ('ping-pong' mode)
- Peripheral indirect addressing
- Null data write mode
- Manual Transfer mode

© 2007 Microchip Technology Incorporated. All Rights Reserved.

University of Microchip UofM MASTERs 2007

DMA Modes: Size and Direction

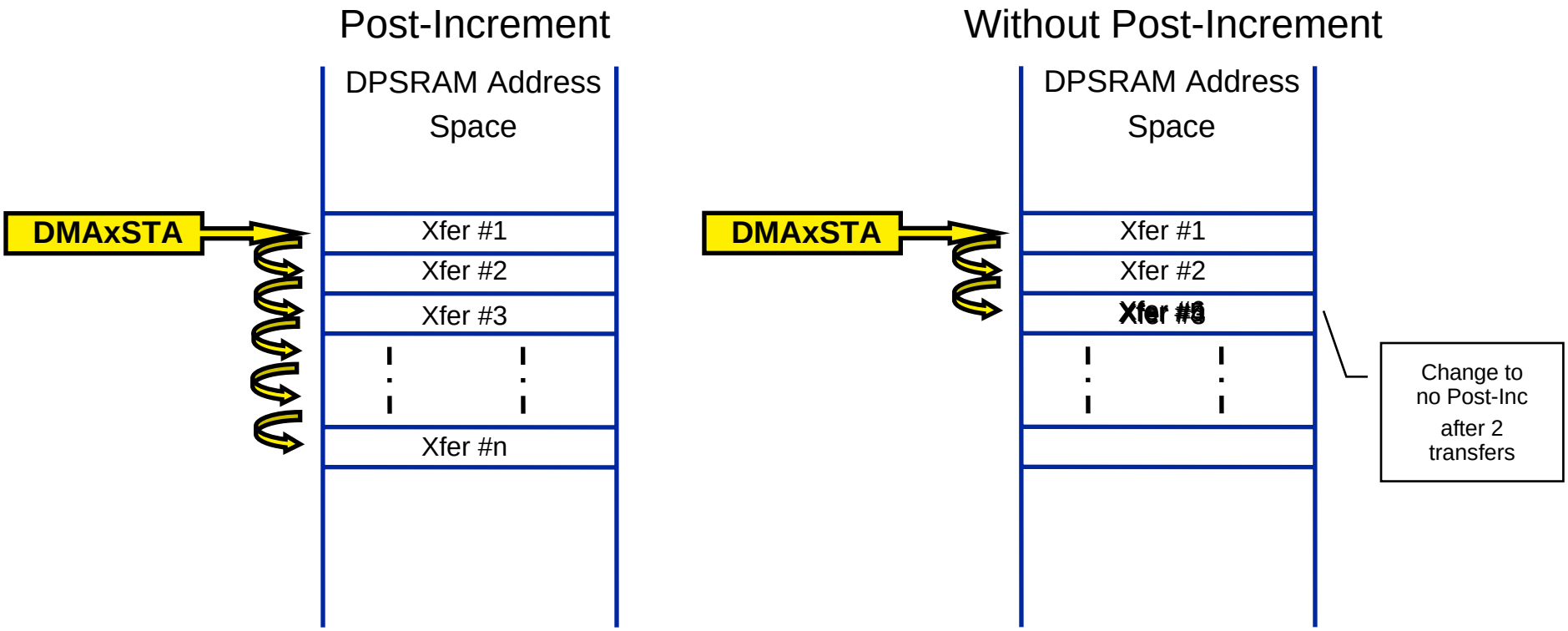
- Word or byte size data transfers
- Peripheral to DPSRAM, or DPSRAM to peripheral transfers

Example: Setup DMA Channel 0 and Channel 1 to transfer words to and from peripheral respectively

```
DMA0CONbits.SIZE = 0;   // Word transfers
DMA0CONbits.DIR  = 1;   // RAM-to-Peripheral direction

DMA1CONbits.SIZE = 0;   // Word transfers
DMA1CONbits.DIR  = 0;   // Peripheral-to-RAM direction
```

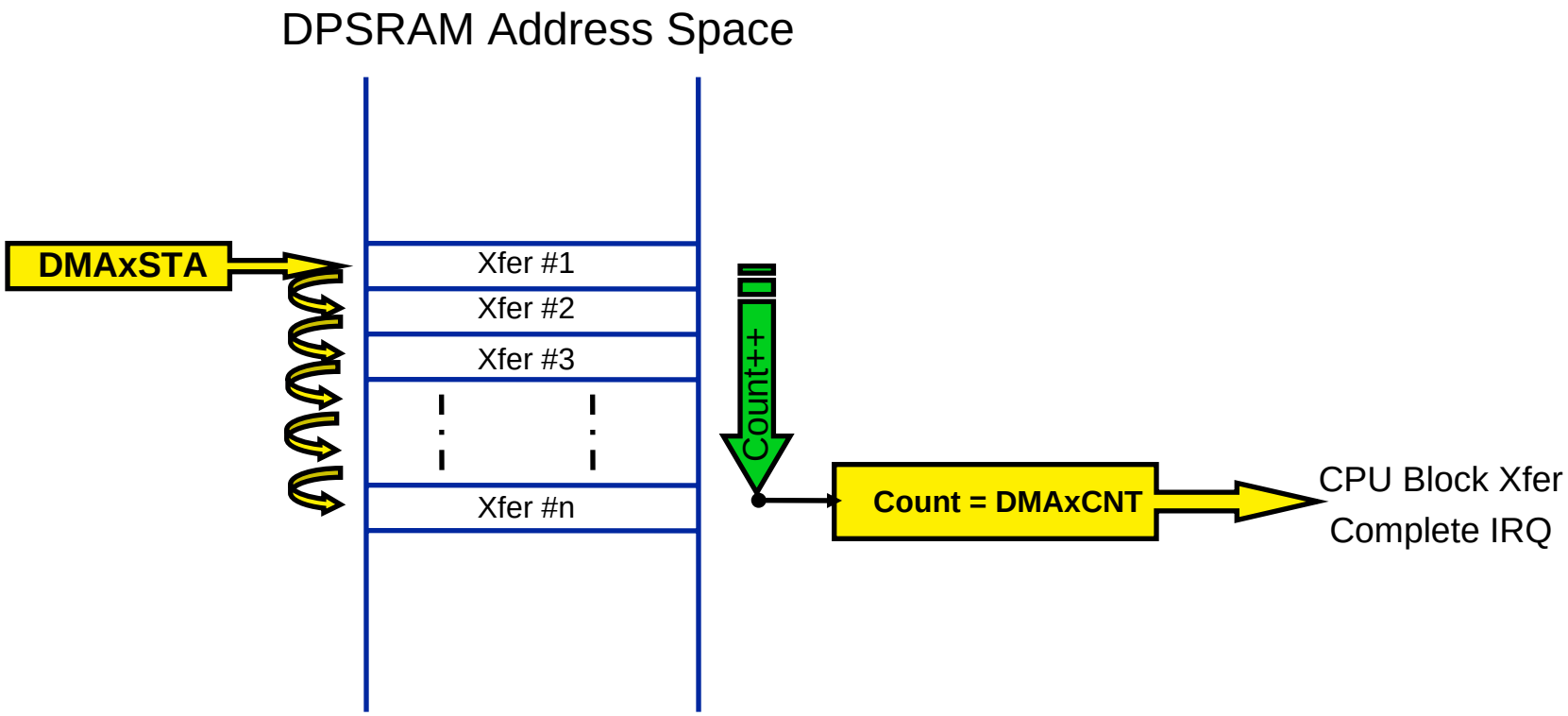
© 2007 Microchip Technology Incorporated. All Rights Reserved.

DMA Modes: Register Indirect Addressing



© 2007 Microchip Technology Incorporated. All Rights Reserved.

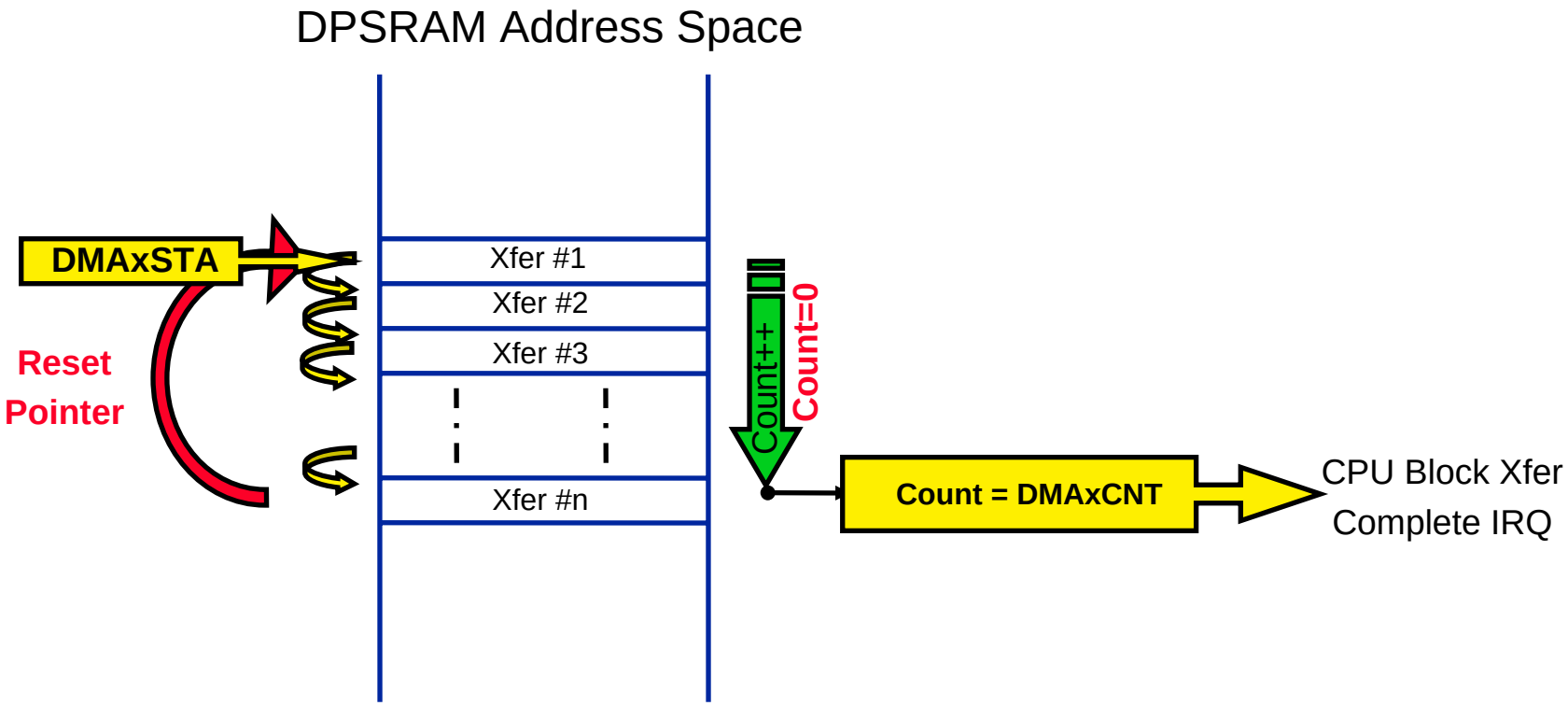
DMA Modes: One-Shot



- Move 1 block of data then disable channel

© 2007 Microchip Technology Incorporated. All Rights Reserved.

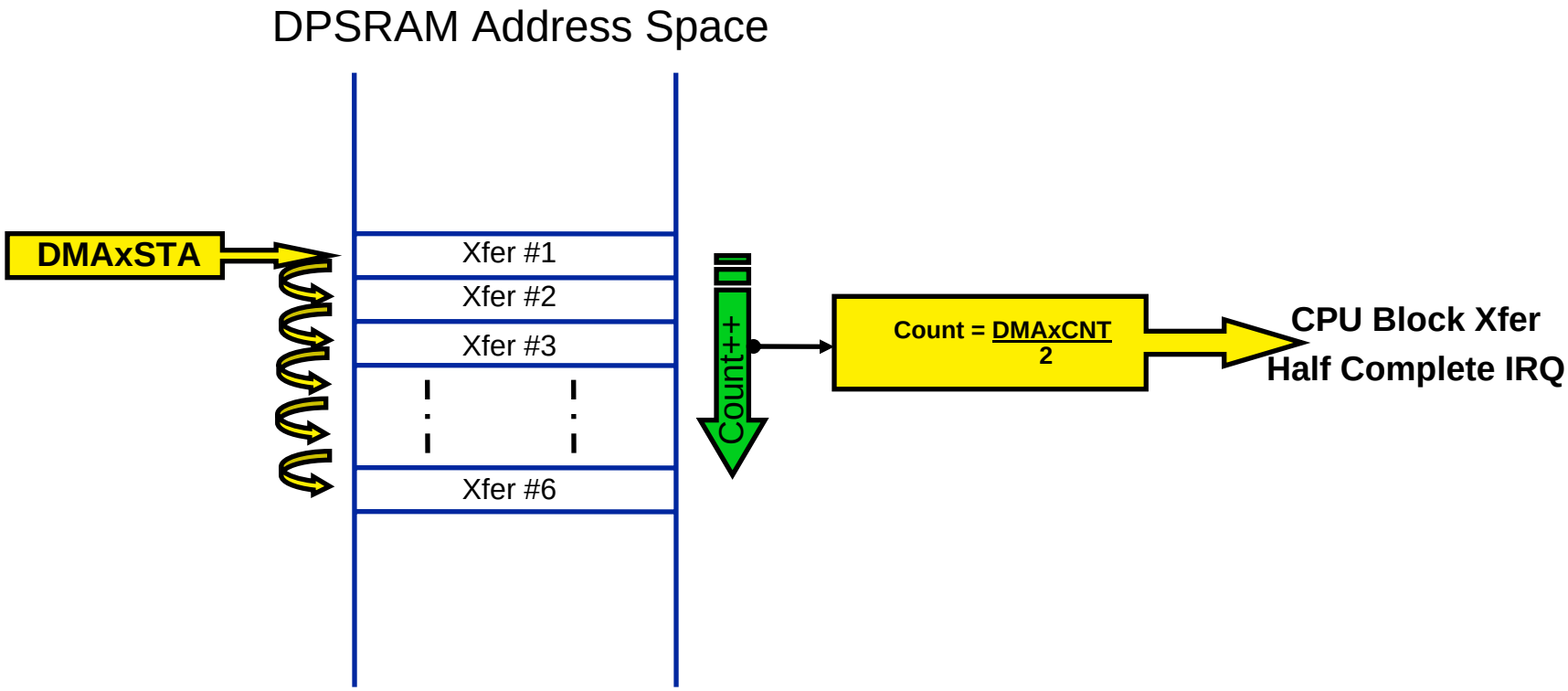
DMA Modes: Continuous



- Move a block of data then automatically configure channel ready to repeat transfer

© 2007 Microchip Technology Incorporated. All Rights Reserved.

DMA Modes: Half or Full Transfer

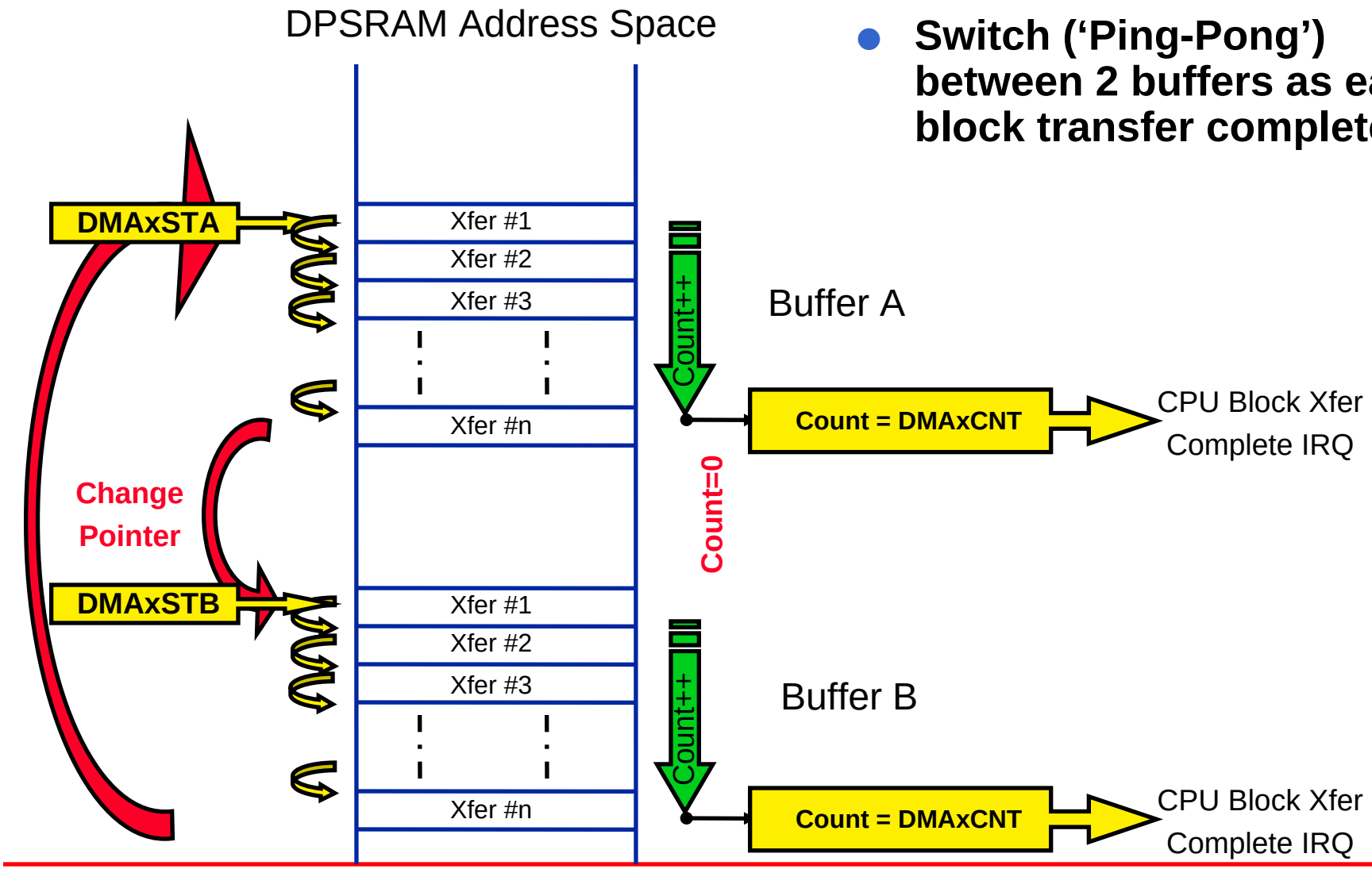


- Move ½ block of data then issue interrupt
- Continue moving second ½ block of data

© 2007 Microchip Technology Incorporated. All Rights Reserved.

DMA Modes: 'Ping-Pong' and Continuous

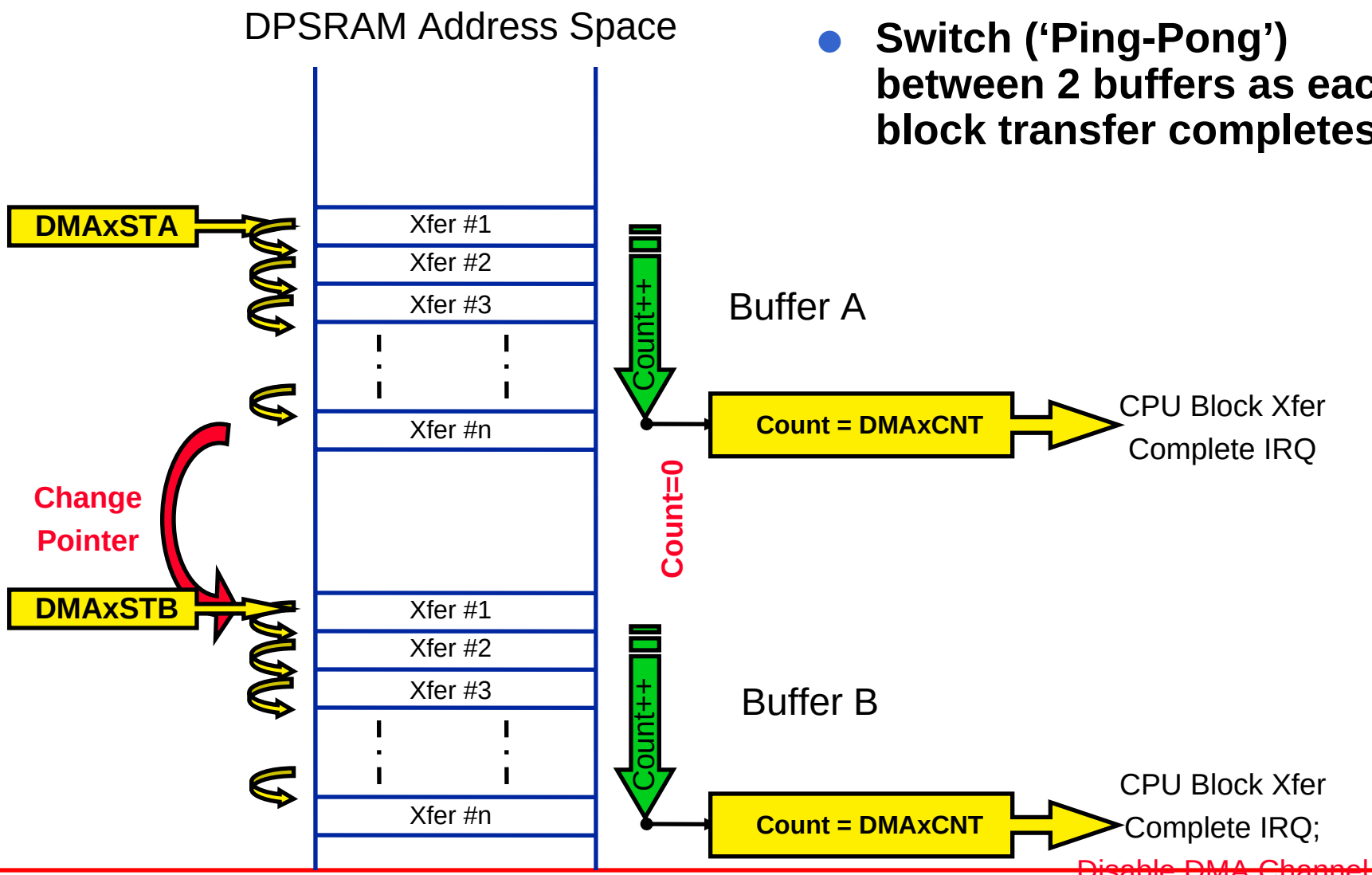
- **Switch ('Ping-Pong') between 2 buffers as each block transfer completes**



© 2007 Microchip Technology Incorporated. All Rights Reserved.

DMA Modes: 'Ping-Pong' and One-Shot

- **Switch ('Ping-Pong') between 2 buffers as each block transfer completes**



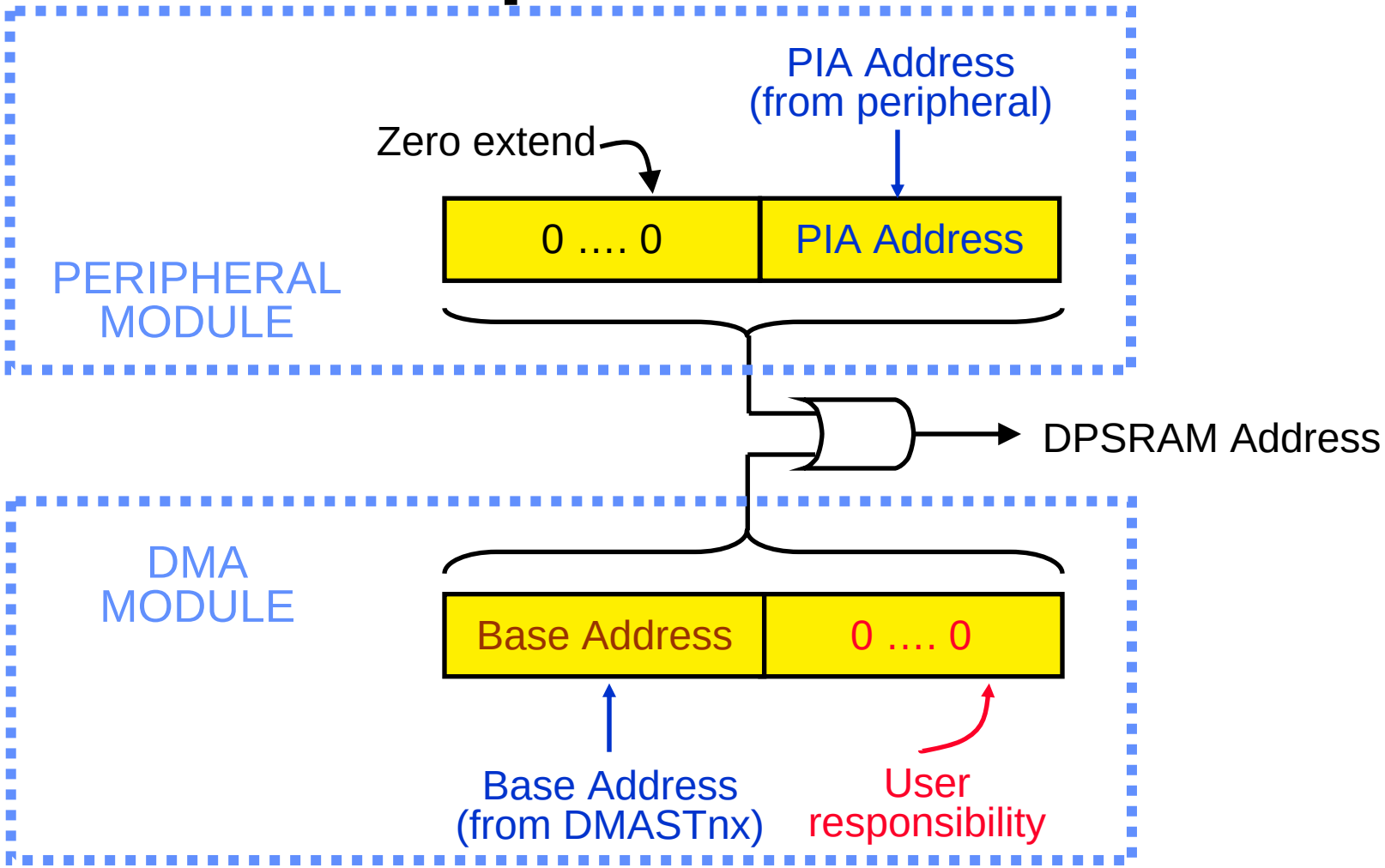
© 2007 Microchip Technology Incorporated. All Rights Reserved.

DMA Modes: Peripheral Indirect

- Least significant bits of address supplied by DMA request peripheral
- Allows ‘scatter/gather’ addressing schemes to be tailored to the needs of each peripheral

© 2007 Microchip Technology Incorporated. All Rights Reserved.

DMA Modes: Peripheral Indirect

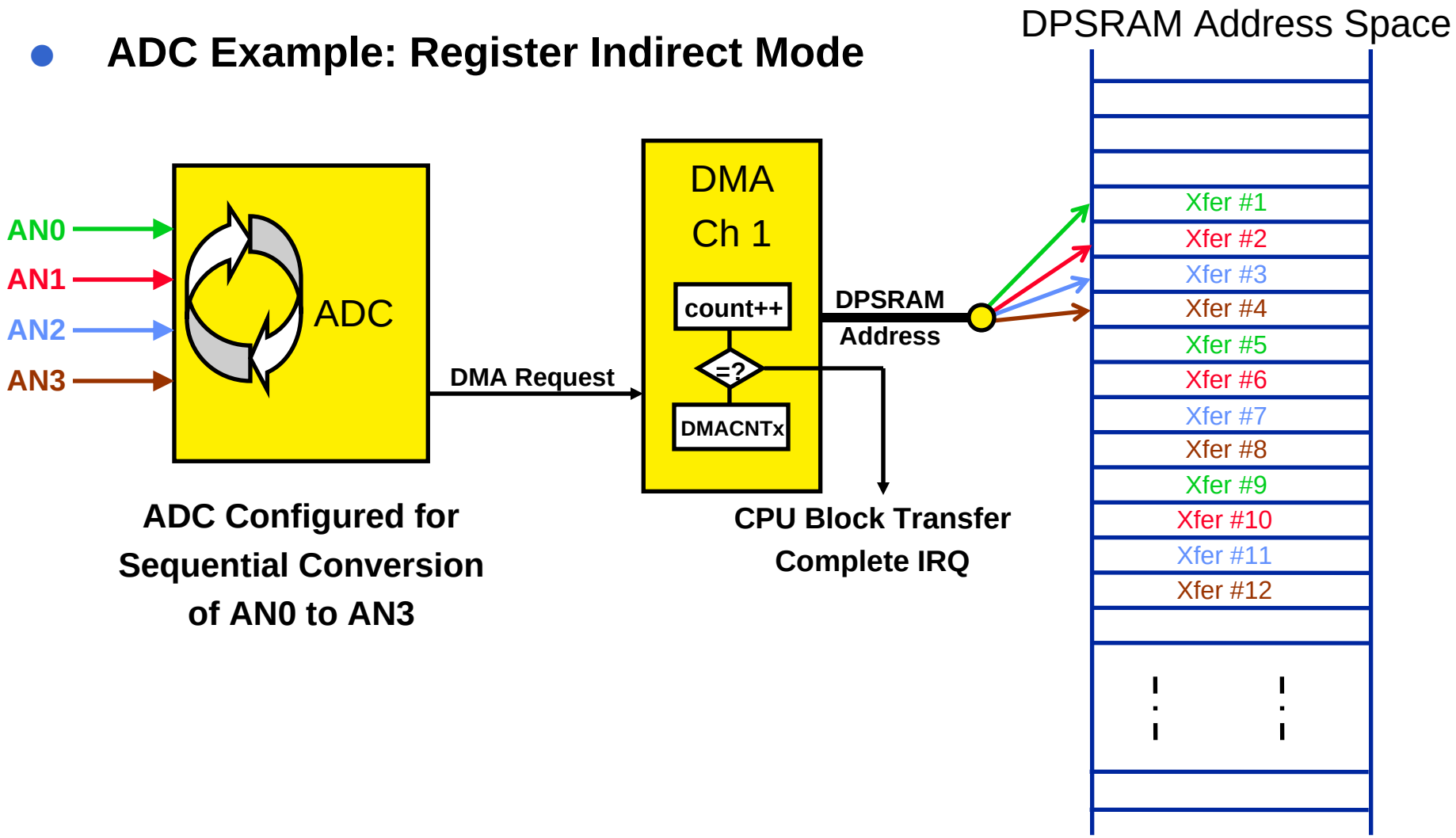


- PIA Mode is also compatible with Auto-Repeat and 'Ping-Pong' modes

© 2007 Microchip Technology Incorporated. All Rights Reserved.

DMA Modes: Peripheral Indirect

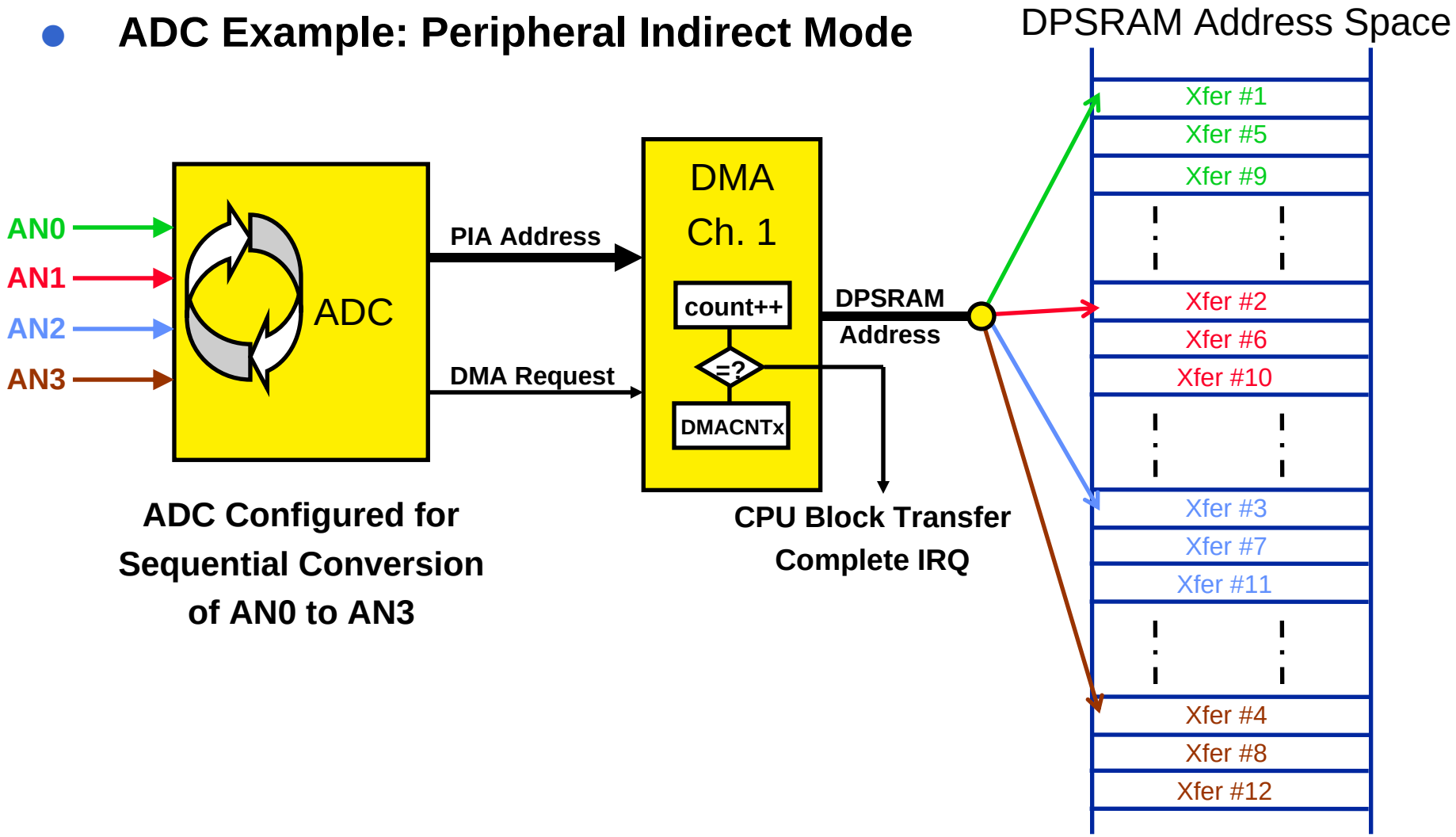
- **ADC Example: Register Indirect Mode**



© 2007 Microchip Technology Incorporated. All Rights Reserved.

DMA Modes: Peripheral Indirect

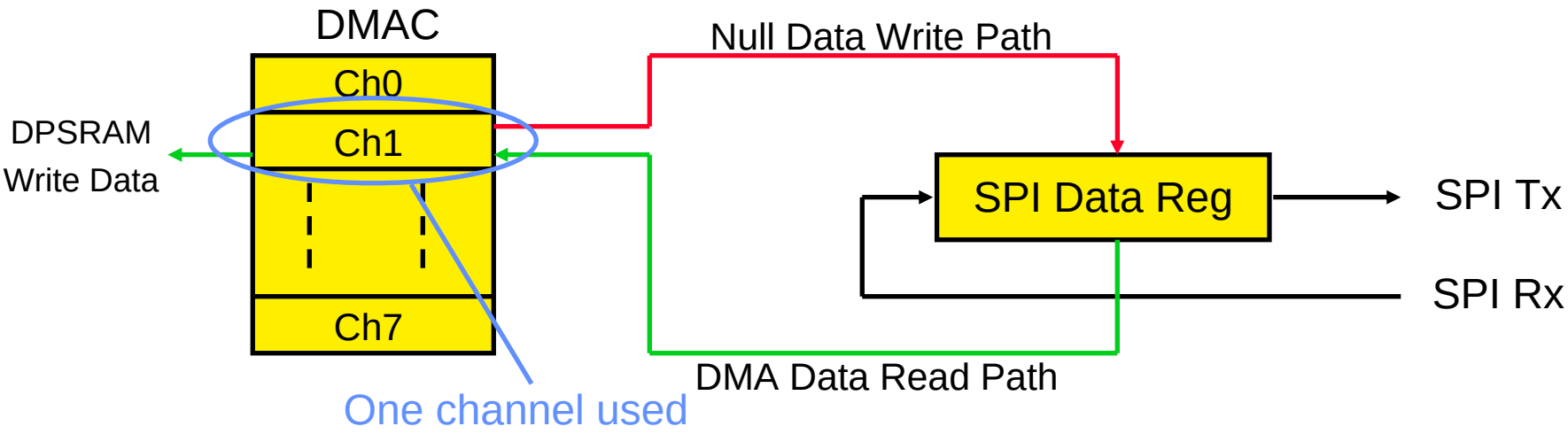
- **ADC Example: Peripheral Indirect Mode**



© 2007 Microchip Technology Incorporated. All Rights Reserved.

DMA Modes: Null Data Write

- **SPI has a single peripheral read/write data address**
  - Requires that data be transmitted (written) in order for external data to be received (read)
  - If only data reception required, “null” (zero) write necessary

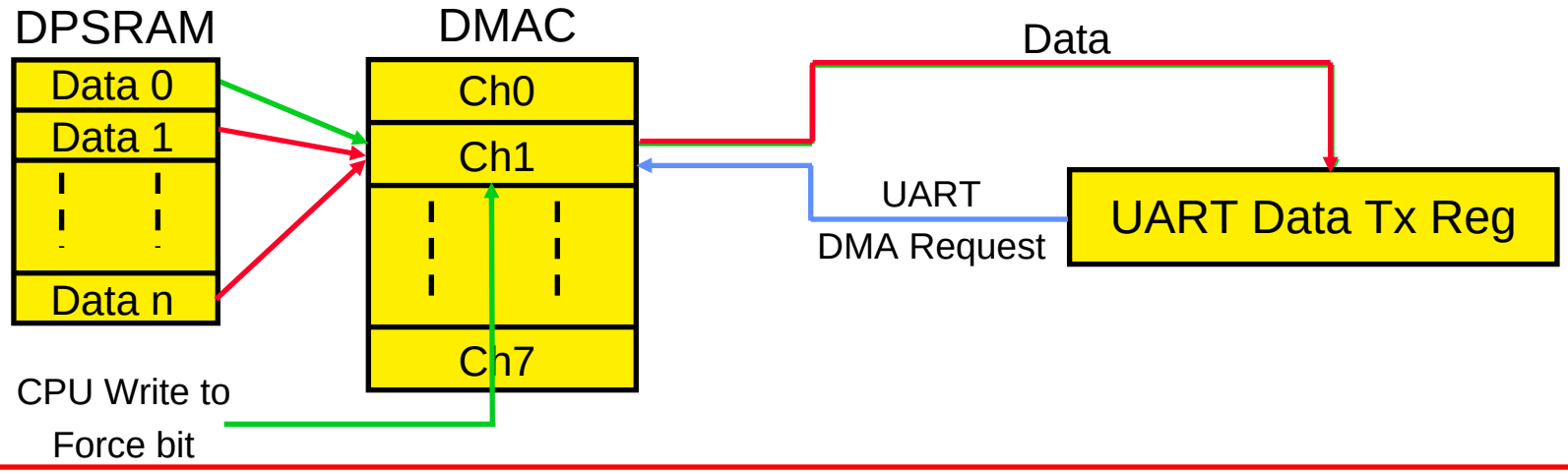


- **“Null write” mode simplifies DMAC operation with SPI**
  - Automatically writes null (zero) data to peripheral data register after each DMA read
  - Avoids wasting another channel for null write

© 2007 Microchip Technology Incorporated. All Rights Reserved.

DMA Modes: Manual Transfer

- **Provides a means to start a DMA transfer using software**
  - Setting the FORCE bit in the selected DMA channel mimics a DMA request
- **Useful for sending the first element from a block of data to a serial peripheral (e.g. UART)**
  - Starts the sequence of DMA request to load data into a peripheral
  - When peripheral data buffer is empty (data sent), peripheral will issue a DMA request for the next data element



© 2007 Microchip Technology Incorporated. All Rights Reserved.

University of Microchip UofM MASTERs 2007

DMA Modes: Examples

Example: Configure DMA Channel 0 for: One-Shot,
Post-Increment,
RAM-to-Peripheral,
Single Buffer

```
DMA0CONbits.AMODE = 0;   // Register Indirect with Post-Increment
DMA0CONbits.MODE  = 1;   // One-Shot, Single Buffer
DMA0CONbits.DIR   = 1;   // RAM-to-Peripheral direction
```

Example: Configure DMA Channel 1 for: Continuous,
Post-Increment,
Peripheral-to-RAM
Ping-Pong

```
DMA1CONbits.AMODE = 0;   // Register Indirect with Post-Increment
DMA1CONbits.MODE  = 2;   // Continuous, Ping-Pong
DMA1CONbits.DIR   = 0;   // Peripheral-to-RAM direction
```

© 2007 Microchip Technology Incorporated. All Rights Reserved.

DMA Interrupts

- **Transfer Complete Interrupt**
- **Write Collision Interrupt (DMA Trap)**
  - Should never happen but if they do, handled robustly
  - CPU will win; DMAC write ignored
  - Cause DMA Fault trap and set channel write collision flag

Example: Enable and process DMA Channel 0 and 1 interrupts

```
IFS0bits.DMA0IF  = 0;              // Clear DMA 0 Interrupt Flag
IEC0bits.DMA0IE  = 1;              // Enable DMA 0 interrupt
IFS0bits.DMA1IF  = 0;              // Clear DMA 1 interrupt
IEC0bits.DMA1IE  = 1;              // Enable DMA 1 interrupt

void __attribute__((__interrupt__)) _DMA0Interrupt(void)
{
      /* Process DMA Channel 0 interrupt here */
      IFS0bits.DMA0IF = 0;         // Clear the DMA0 Interrupt Flag
}
void __attribute__((__interrupt__)) _DMA1Interrupt(void)
{
      /* Process DMA Channel 1 interrupt here */
      IFS0bits.DMA1IF = 0;         // Clear the DMA1 Interrupt Flag
}
```

© 2007 Microchip Technology Incorporated. All Rights Reserved.

DMA Controller
Debug Support

- **2 DMA debug assist registers included**
  - DSADR<15:0> : Captures the DPSRAM address of the most recent DMA transfer
  - DMACS1 :
    - **LSTCH<2:0> : Captures the ID of the most recently active DMA channel**
    - **PPSTx : 'Ping-Pong' mode status bits, one per channel.**
      Indicates which buffer is active (A or B)

© 2007 Microchip Technology Incorporated. All Rights Reserved.

10-bit A/D Converter

High Speed Analog Feedback

© 2007 Microchip Technology Incorporated. All Rights Reserved.

A/D Feature Summary

- 10-bit Resolution with +/- 1-bit accuracy
- Up to 1.1Msps
- Up to 32 analog inputs, 4 S/H Amplifiers

© 2007 Microchip Technology Incorporated. All Rights Reserved.

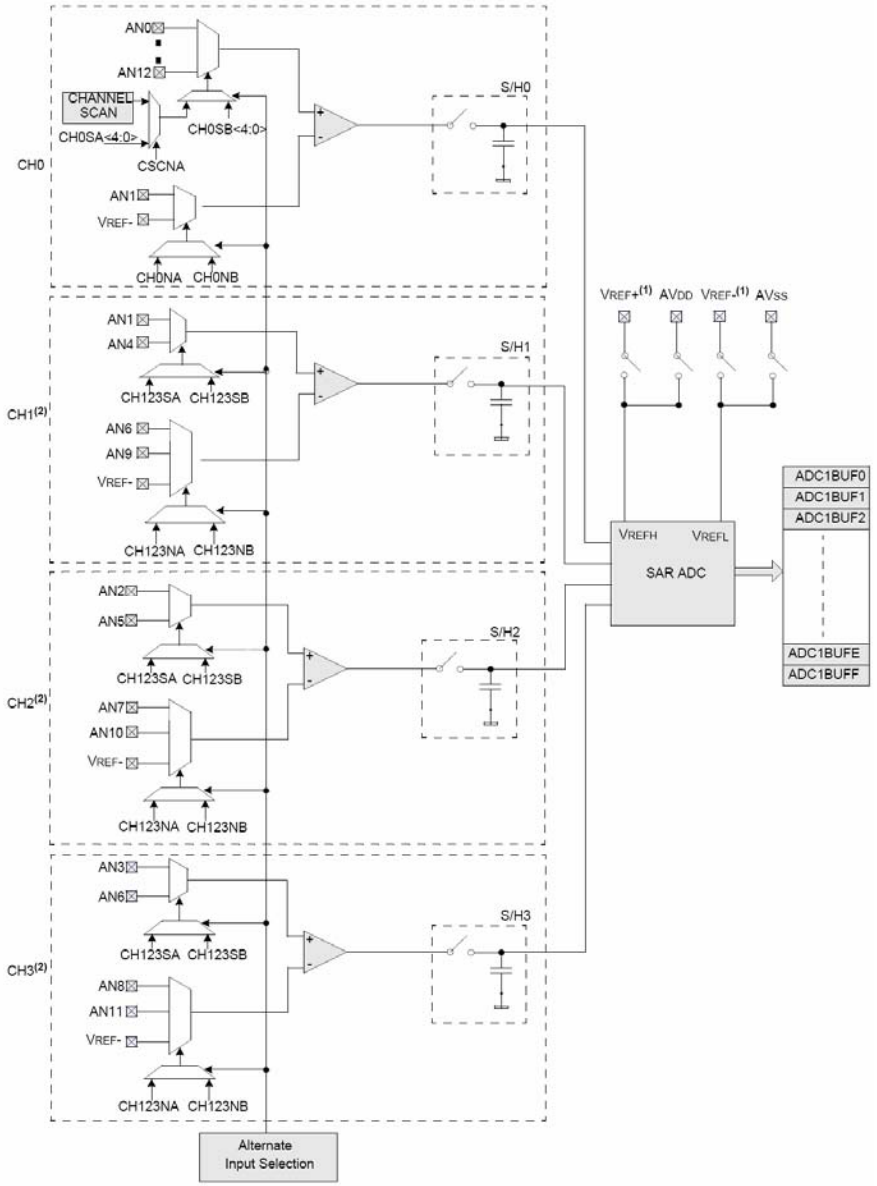
University of Microchip UofM MASTERs 2007

A/D Feature Summary

- **External VREF+ and VREF-**
- **Programmable sampling sequence**
  - Result Buffers
  - Scan mode
  - Alternate sample mode
- **Multiple conversion trigger sources**
- **Selectable result formats**
- **Conversions in Sleep and Idle**

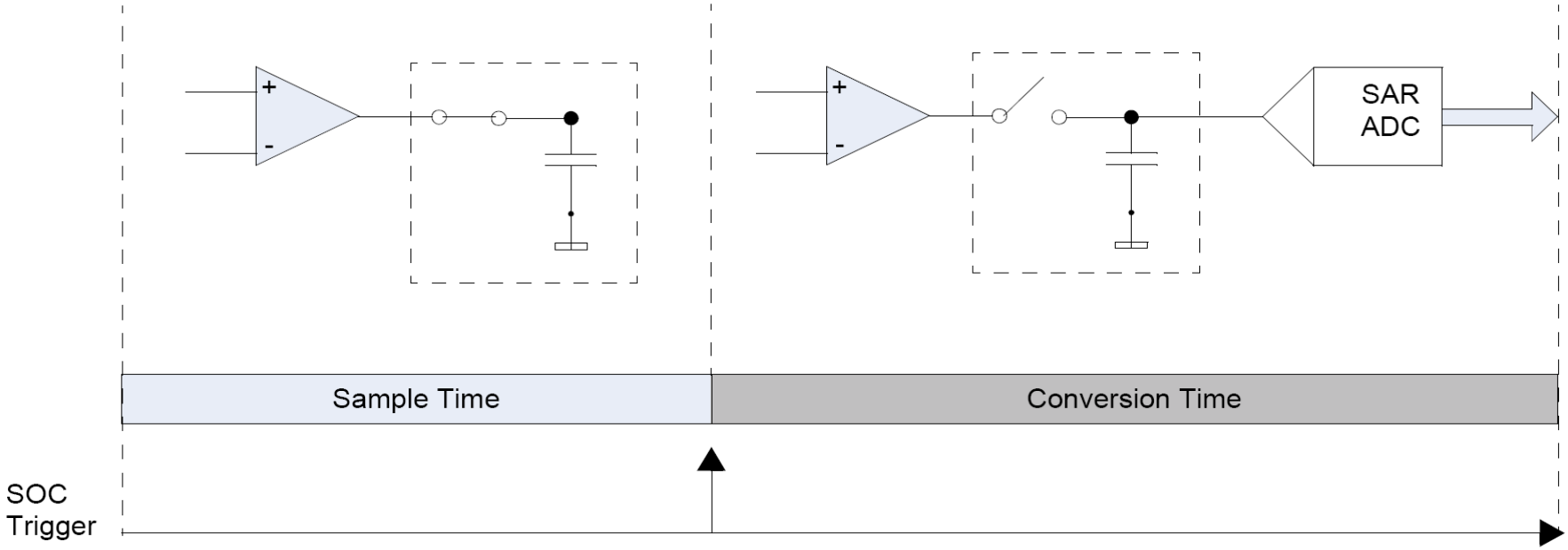
© 2007 Microchip Technology Incorporated. All Rights Reserved.

10-bit A/D Block Diagram



© 2007 Microchip Technology Incorporated. All Rights Reserved.

Sample/Conversion Sequence



© 2007 Microchip Technology Incorporated. All Rights Reserved.

Conversion Time

- **10-bit ADC Conversion Time**

  Conversion Time = 12 * T_{AD}

  Where : T_{AD} = ADC Clock Period

- **12-bit ADC Conversion Time**

  Conversion Time = 14 * T_{AD}

  Where : T_{AD} = ADC Clock Period

© 2007 Microchip Technology Incorporated. All Rights Reserved.

Start of Conversion (SOC) Trigger Selection

SSRC <2:0>	SOC Trigger Source
000	Manual Trigger
001	External Interrupt Trigger
010	Timer Interrupt Trigger
011	Motor Control PWM
100	Reserved
101	Reserved
110	Reserved
111	Automatic Trigger

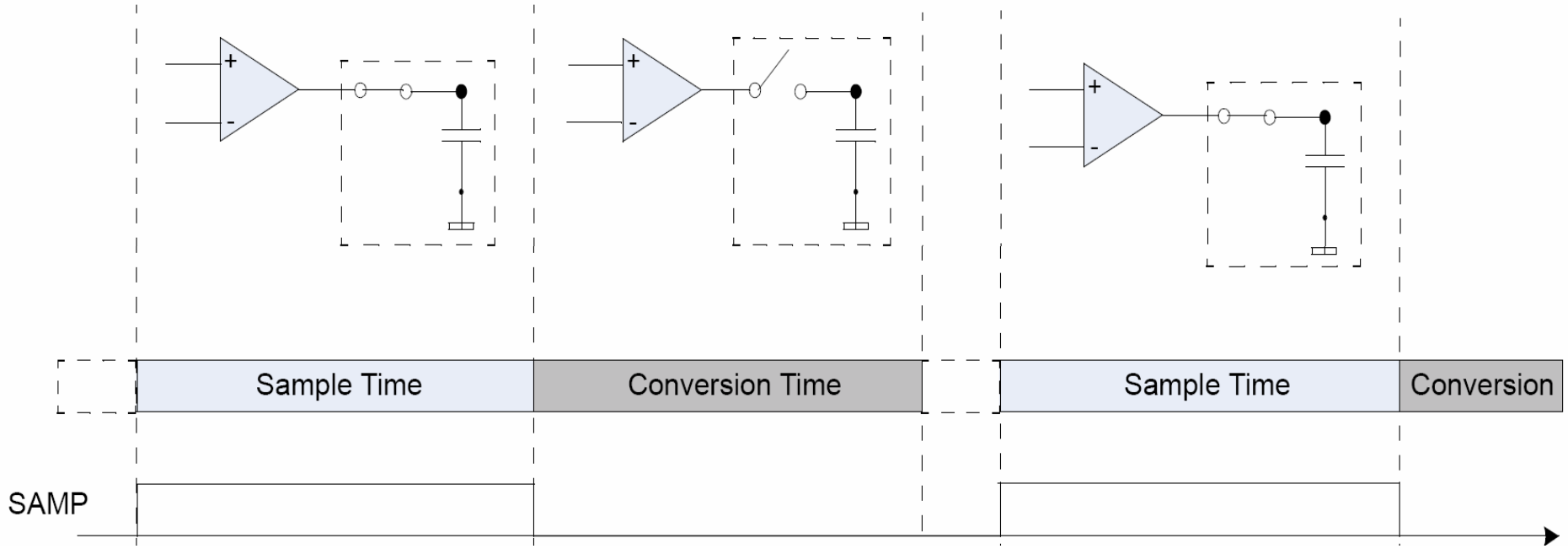
© 2007 Microchip Technology Incorporated. All Rights Reserved.

Sample/Conversion Sequence

ASAM	SSRC <2:0>	Description
0	000	Manual Sample and Manual Conversion
0	111	Manual Sample and Automatic Conversion
0 0 0	001 010 011	Manual Sample and Triggered Conversion
1	000	Automatic Sample and Manual Conversion
1	111	Automatic Sample and Automatic Conversion
1 1 1	001 101 011	Automatic Sample and Triggered Conversion

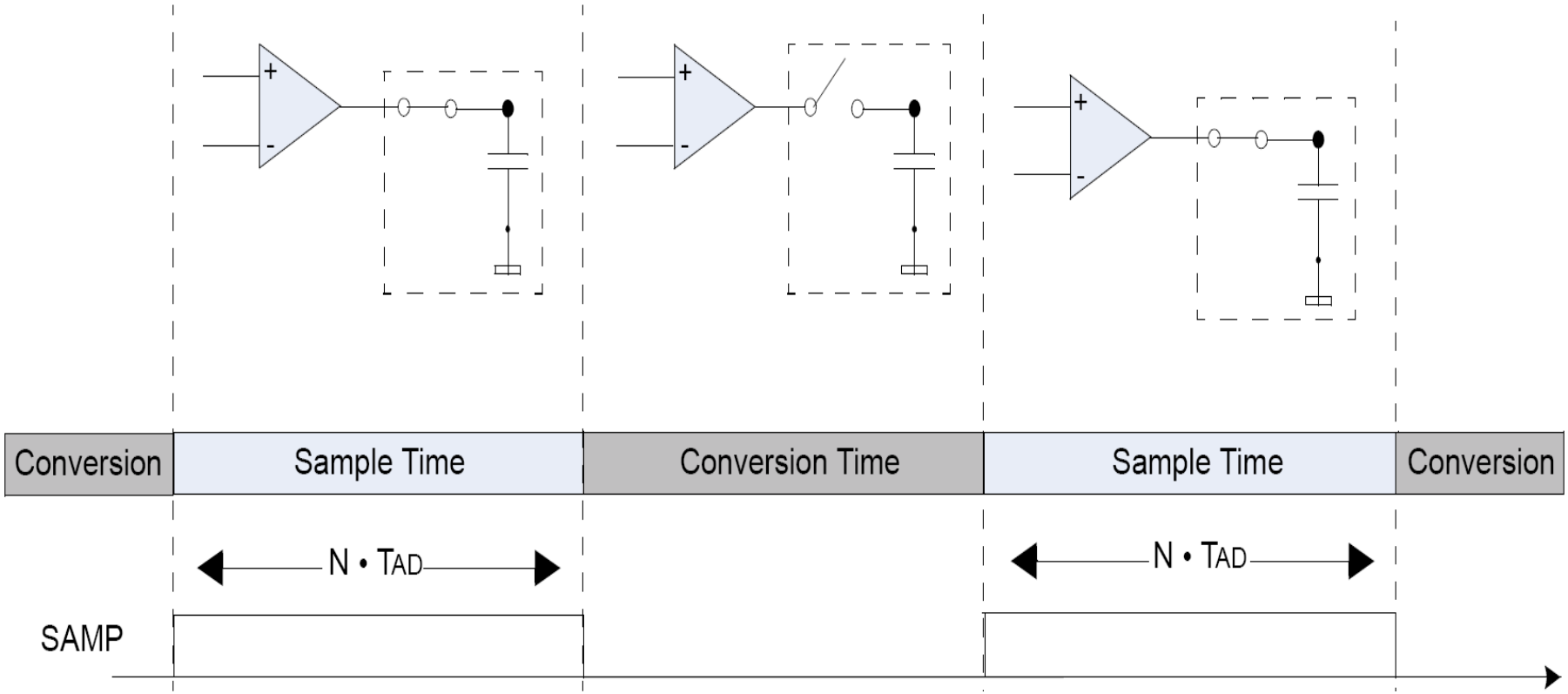
© 2007 Microchip Technology Incorporated. All Rights Reserved.

Manual Sample and Manual Conversion Sequence



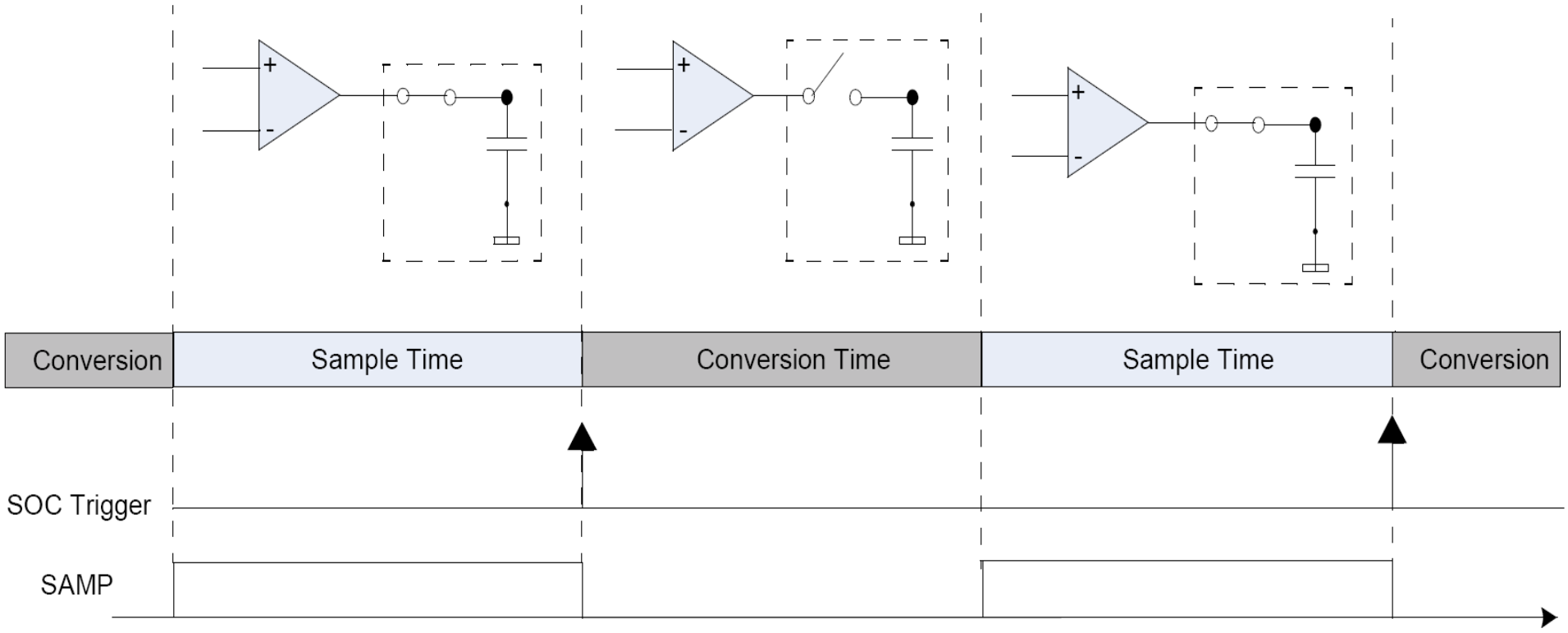
© 2007 Microchip Technology Incorporated. All Rights Reserved.

Automatic Sample and Automatic Conversion Sequence



© 2007 Microchip Technology Incorporated. All Rights Reserved.

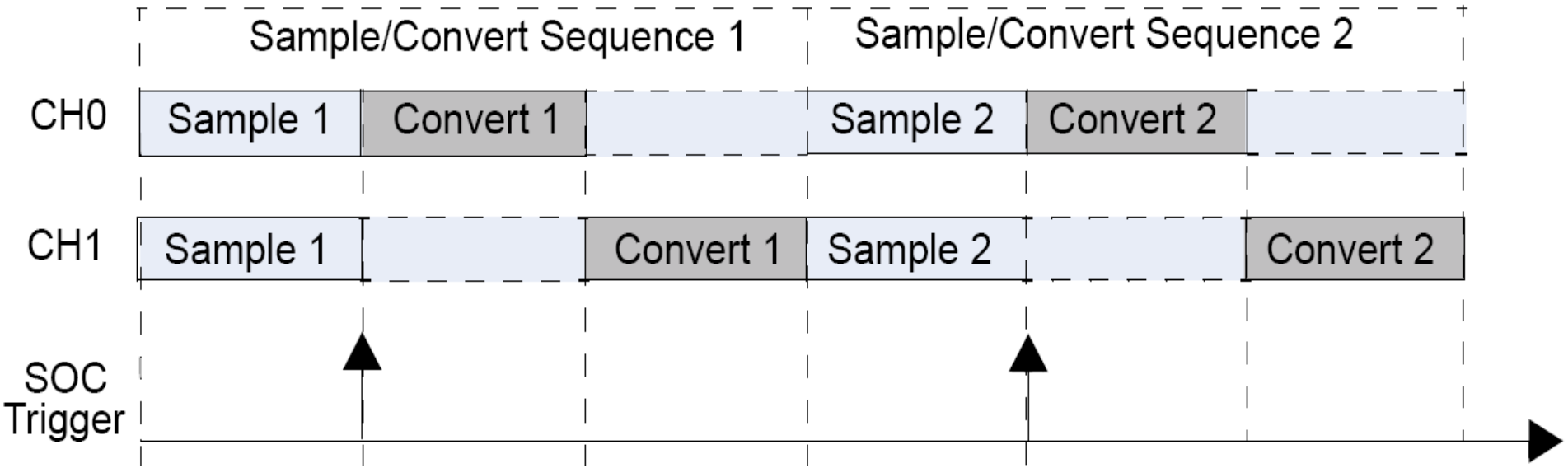
Automatic Sample and Triggered Conversion Sequence



© 2007 Microchip Technology Incorporated. All Rights Reserved.

Multi-Channel Sample Conversion Sequence

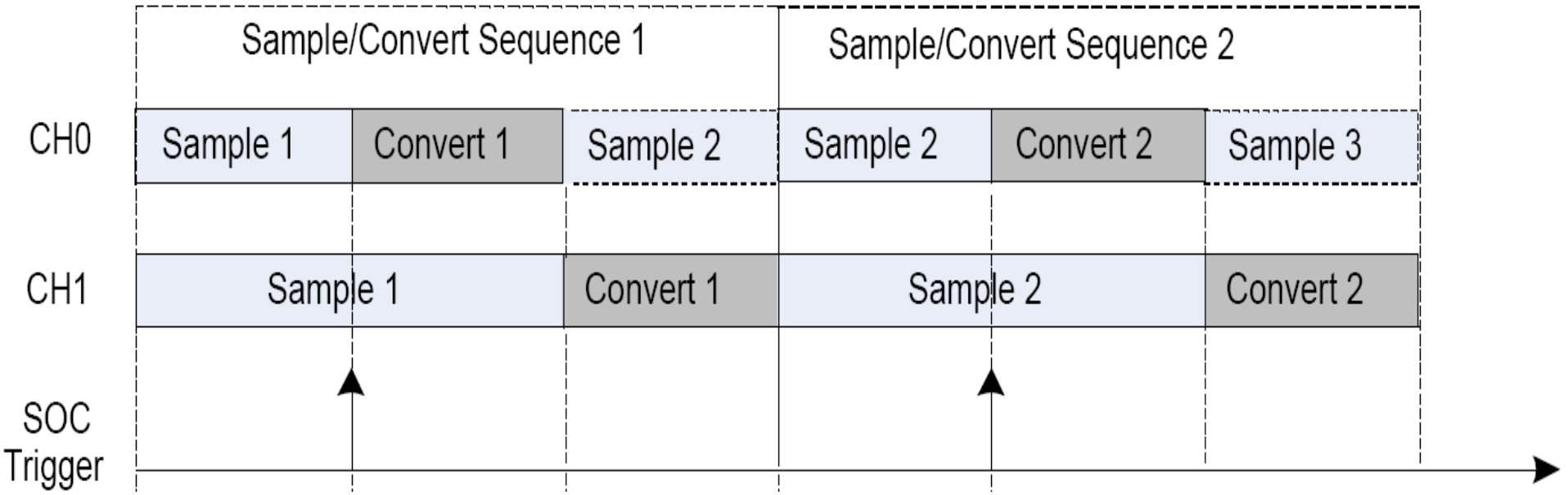
- **2-Channel Simultaneous Sampling**



© 2007 Microchip Technology Incorporated. All Rights Reserved.

Multi-Channel Sample Conversion Sequence (Cont.)

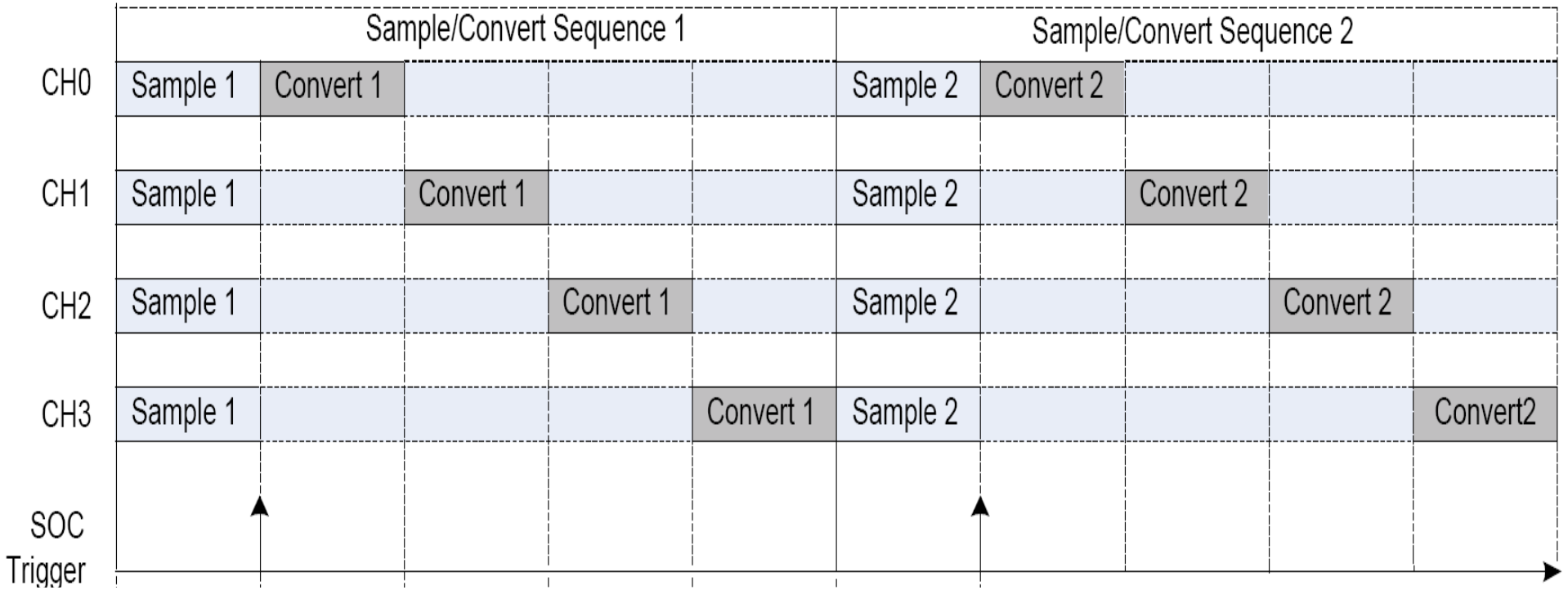
- **2-Channel Sequential Sampling**



© 2007 Microchip Technology Incorporated. All Rights Reserved.

Multi-Channel Sample Conversion Sequence (Cont.)

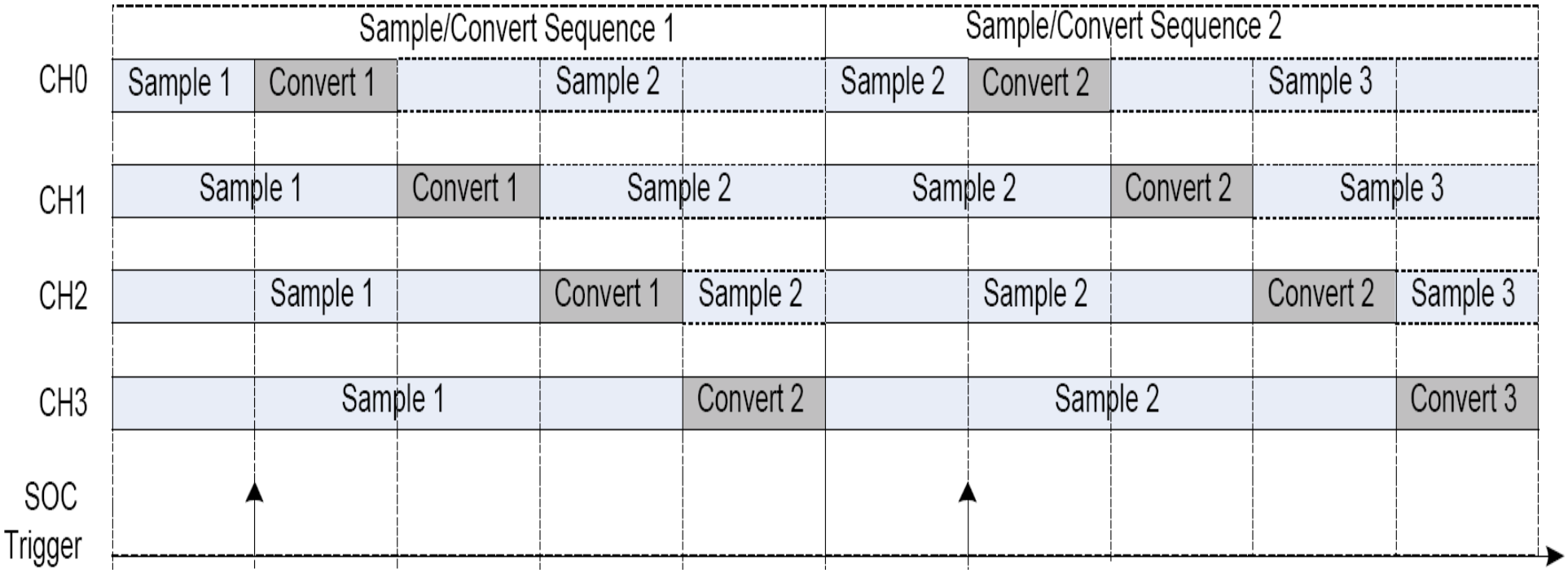
- **4-Channel Simultaneous Sampling**



© 2007 Microchip Technology Incorporated. All Rights Reserved.

Multi-Channel Sample Conversion Sequence (Cont.)

- **4-Channel Sequential Sampling**



© 2007 Microchip Technology Incorporated. All Rights Reserved.

ADC Operational Mode

- **10-bit, 4- Channel Mode (AD12B = 0)**
  - Dual Channel (CH0, CH1)
  - Multi-Channel (CH0 – CH3)

- **12-bit, Single Channel (AD12B = 1)**
  - Single Channel (CH0)

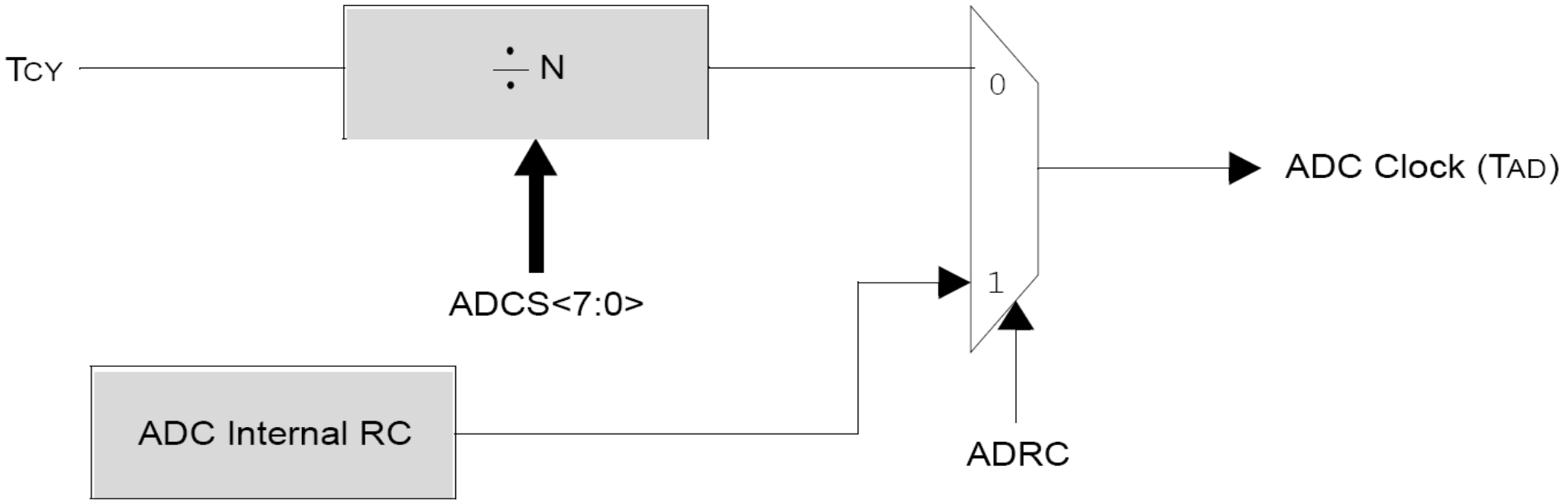
© 2007 Microchip Technology Incorporated. All Rights Reserved.

ADC Clock Selection

- **ADC Clock Period**

  ADC Clock Period (T_{AD}) = T_{CY} * (ADCS + 1)

- **ADC Clock Generation**



© 2007 Microchip Technology Incorporated. All Rights Reserved.

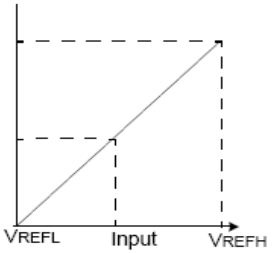
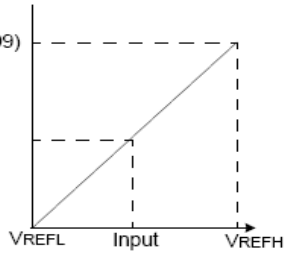
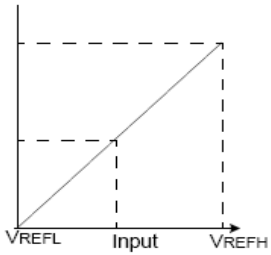
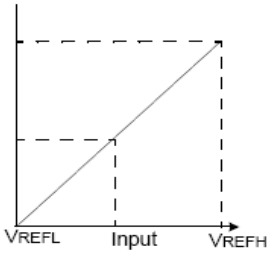
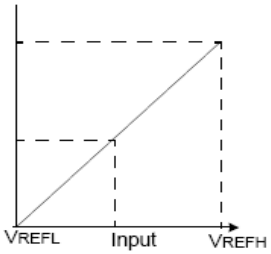
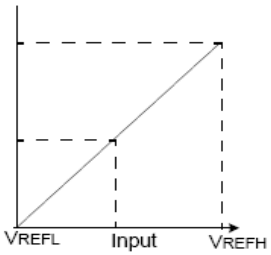
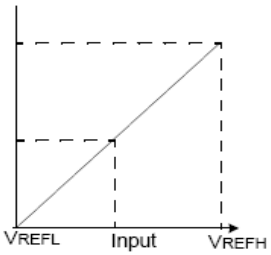
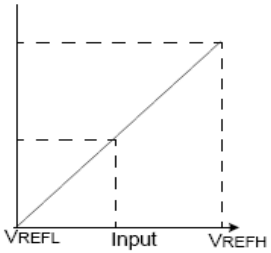
UNIVERSITY OF MICROCHIP UOFM MASTERS 2007

Output Data Format

- Signed Fractional
- Unsigned Fractional
- Signed Integer
- Unsigned Integer

© 2007 Microchip Technology Incorporated. All Rights Reserved.

Output Data Format (Cont.)

	10-bit ADC	12-bit ADC
FORM = 0b11 Signed Fraction(Q15)	0111 1111 1100 0000 (+0.999) 0000 0000 0000 0000 (0) 1000 0000 0000 0000 (-1) 	0111 1111 1111 0000 (+0.999) 0000 0000 0000 0000 (0) 1000 0000 0000 0000 (-1) 
FORM = 0b10 Unsigned Fraction (Q16)	1111 1111 1100 0000 (+0.999) 1000 0000 0000 0000 (0.5) 0000 0000 0000 0000 (0) 	1111 1111 1111 0000 (+0.999) 1000 0000 0000 0000 (0.5) 0000 0000 0000 0000 (0) 
FORM = 0b01 Signed Integer	0000 0001 1111 1111 (511) 0000 0000 0000 0000 (0) 1111 1110 0000 0000 (-512) 	0000 0111 1111 1101 (2045) 0000 0000 0000 0000 (0) 1111 1000 0000 0010 (-2046) 
FORM = 0b00 Unsigned Integer	0000 0011 1111 1111 (1023) 0000 0010 0000 0000 (512) 0000 0000 0000 0000 (0) 	0000 0011 1111 1111 (4095) 0000 0010 0000 0000 (2048) 0000 0000 0000 0000 (0) 

© 2007 Microchip Technology Incorporated. All Rights Reserved.

ADC Interrupt Generation

- **Conversions per ADC Interrupt Depend On:**
  - Number of Sample/Hold Channels Selected
  - Sequential or Simultaneous Sampling
  - Samples Convert Sequences Per Interrupt SMPI <3:0>

© 2007 Microchip Technology Incorporated. All Rights Reserved.

ADC Interrupt Generation (No DMA)

CHPS <1:0>	SIMSAM	SMPI <3:0>	Conversions/ Interrupt	Description
00	X	N-1	N	1-Channel Mode
01	0	N-1	N	2-Channel Sequential Sampling
1x	0	N-1	N	4-Channel Sequential Sampling
01	1	N-1	2N	2-Channel Simultaneous Sampling
1x	1	N-1	4N	4-Channel Simultaneous Sampling

© 2007 Microchip Technology Incorporated. All Rights Reserved.

Analog Input Selection for Conversion

- Fixed Input Selection
- Alternate Input Selection
- Channel Scanning (CH0)

© 2007 Microchip Technology Incorporated. All Rights Reserved.

Analog Input Selection for Conversion (Cont.)

- **Fixed Input Selection**

© 2007 Microchip Technology Incorporated. All Rights Reserved.

Analog Input Selection for Conversion (Cont.)

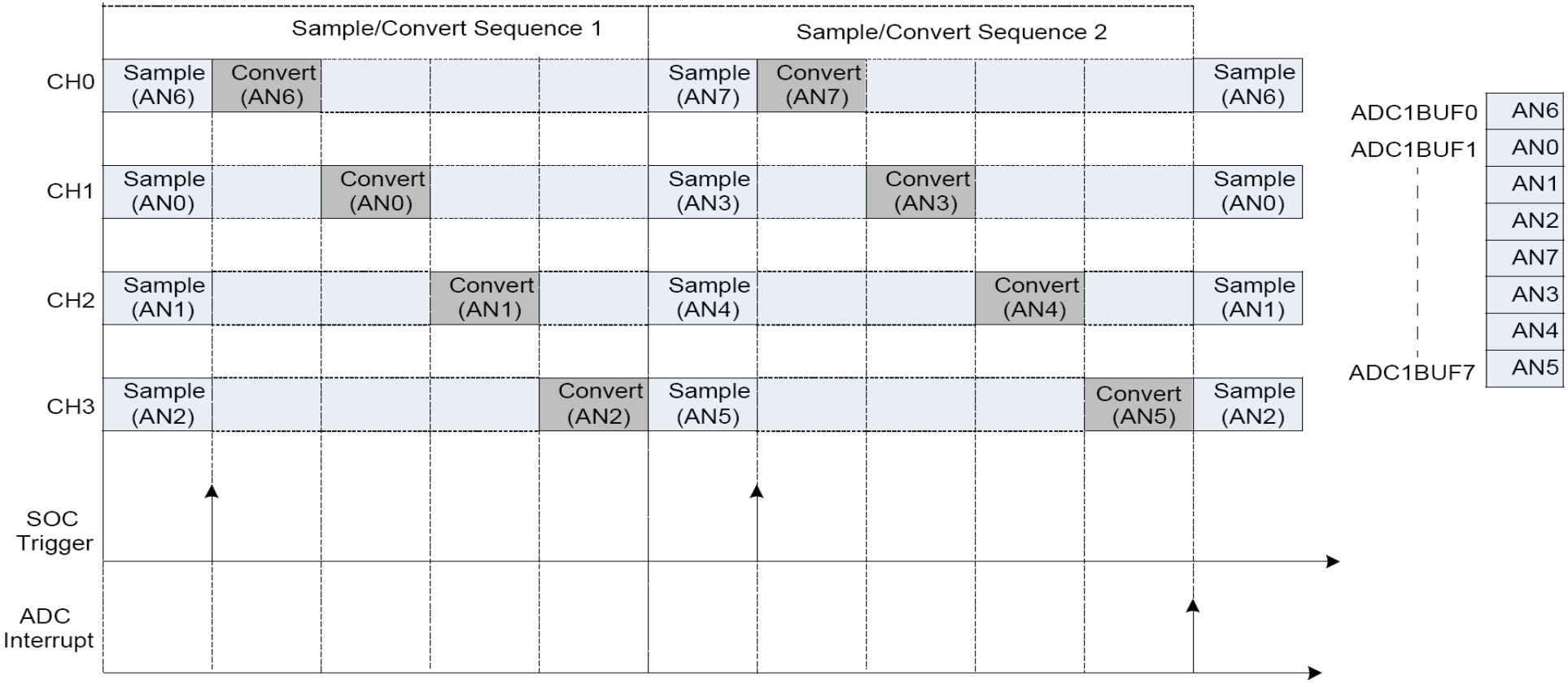
- **Alternate Input Selection**

CHPS <1:0>	SIMSAM	SMPI <3:0>	Conversions/ Interrupt	Description
00	X	1,3,5,7,9,11, 13,15	2,4,6,8,10,12, 14,16	1- Channel Mode
01	0	3,7,11,15	4,8,12,16	2- Channel Sequential Sampling
1x	0	7,15	8,16	4- Channel Sequential Sampling
01	1	1,3,5,7	4,8,12,16	2- Channel Simultaneous Sampling
1x	1	1,3	8,16	4- Channel Simultaneous Sampling

© 2007 Microchip Technology Incorporated. All Rights Reserved.

Analog Input Selection for Conversion (Cont.)

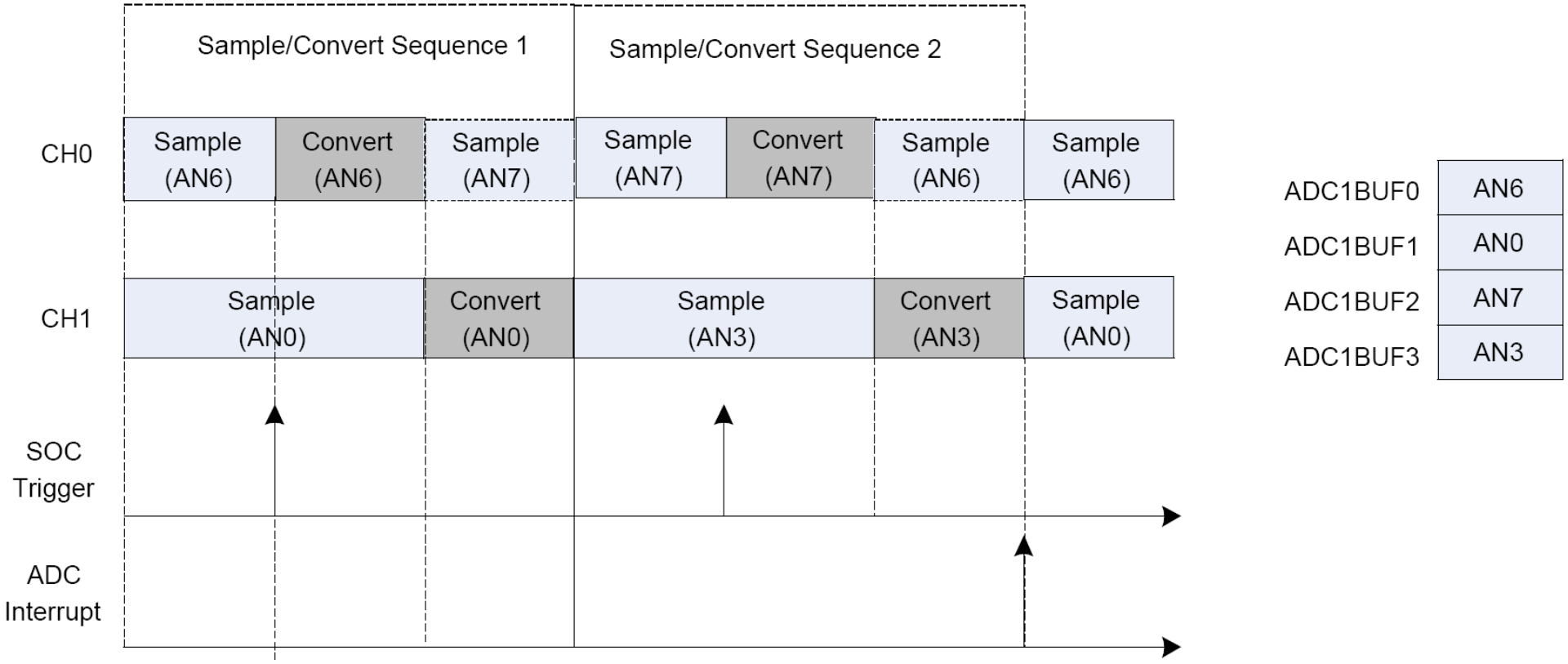
- **Alternative Input Selection in 4-Channel Simultaneous Sampling Mode**



© 2007 Microchip Technology Incorporated. All Rights Reserved.

Analog Input Selection for Conversion (Cont.)

- **Alternative Input Selection in 2-Channel Sequential Sampling Mode**



© 2007 Microchip Technology Incorporated. All Rights Reserved.

Analog Input Selection for Conversion (Cont.)

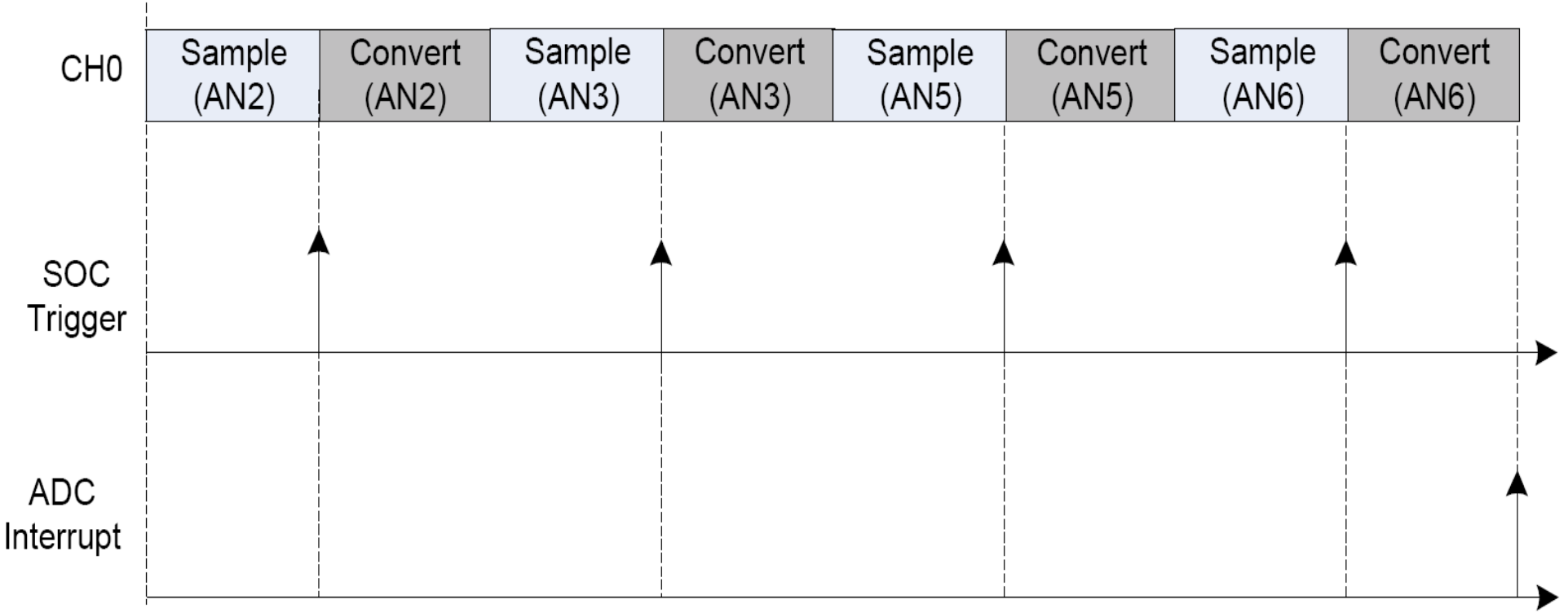
- **Channel Scanning**

CHPS <1:0>	SIMSAM	SMPI <3:0>	Conversions/ Interrupt	Description
00	X	N-1	N	1-Channel Mode
01	0	2N-1	2N	2-Channel Sequential Sampling
1x	0	4N-1	4N	4-Channel Sequential Sampling
01	1	N-1	2N	2-Channel Simultaneous Sampling
1x	1	N-1	4N	4-Channel Simultaneous Sampling

© 2007 Microchip Technology Incorporated. All Rights Reserved.

Analog Input Selection for Conversion (Cont.)

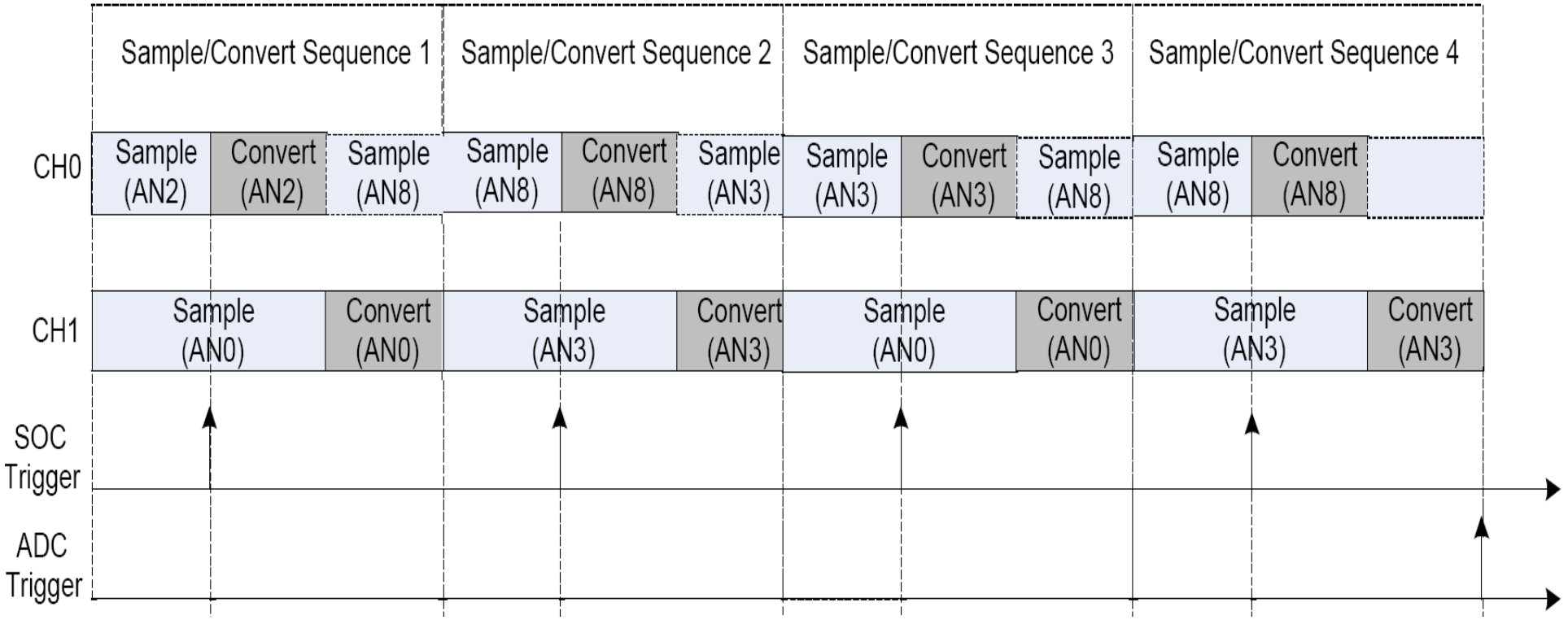
- **Scanning Four Analog Inputs Using CH0**



© 2007 Microchip Technology Incorporated. All Rights Reserved.

Analog Input Selection for Conversion (Cont.)

- **Channel Scan With Alternate Input Selection**



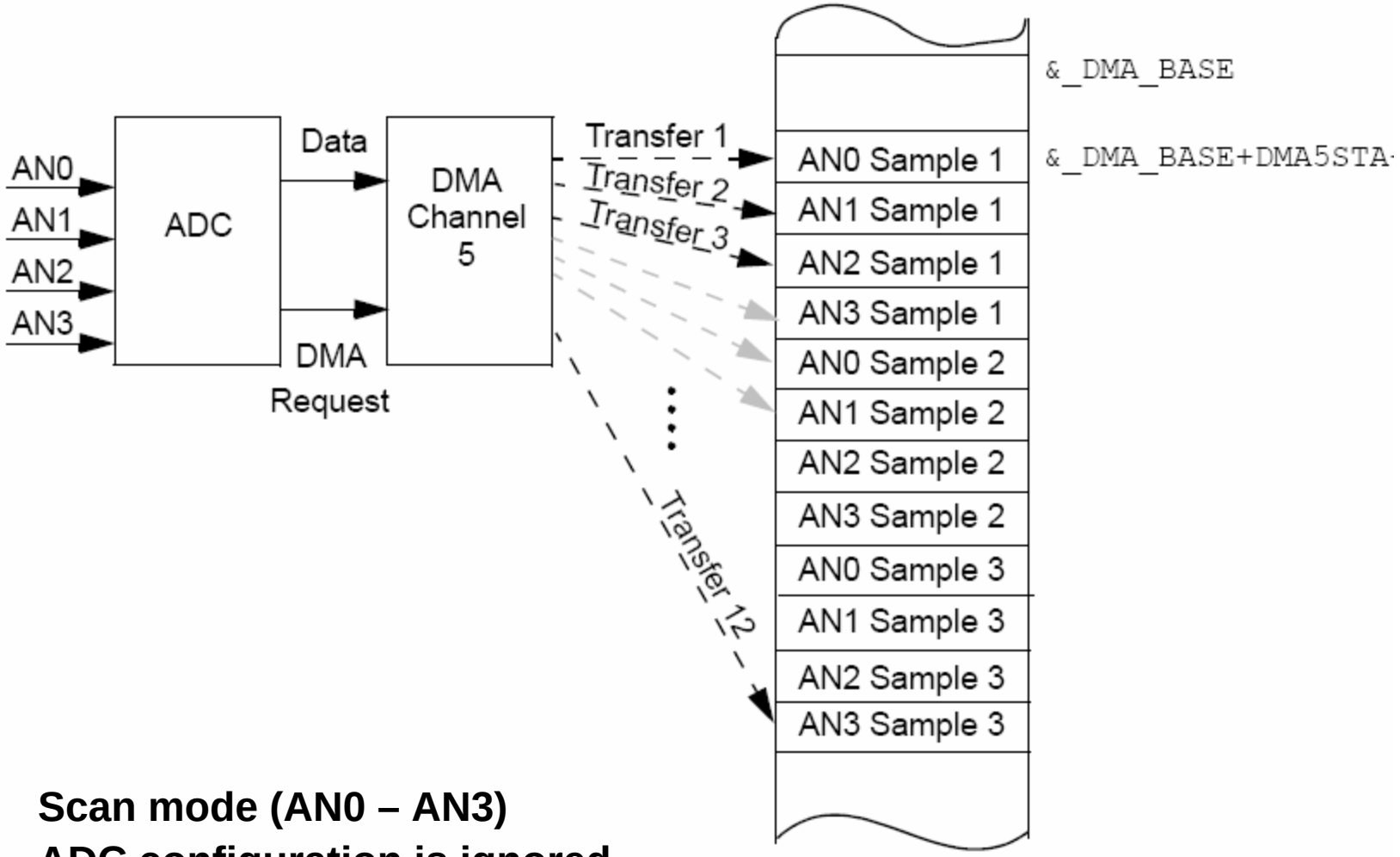
© 2007 Microchip Technology Incorporated. All Rights Reserved.

ADC with DMA Configuration

- Register Indirect
- Peripheral Indirect

© 2007 Microchip Technology Incorporated. All Rights Reserved.

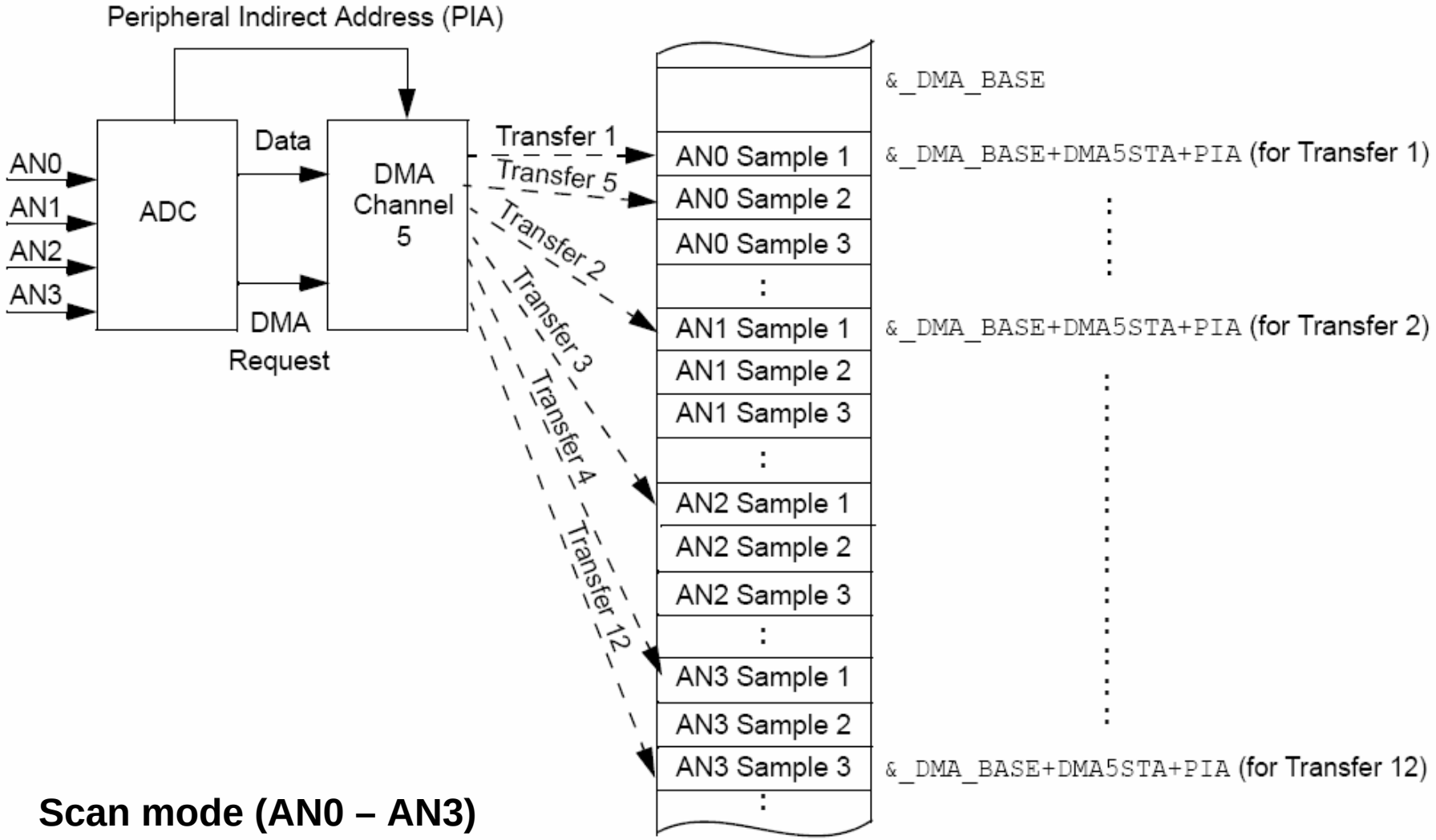
DMA Transfers – Register Indirect (Order of Conversion)



- Scan mode (AN0 – AN3)
- ADC configuration is ignored
- DMA count is used to generate DMA Interrupt

© 2007 Microchip Technology Incorporated. All Rights Reserved.

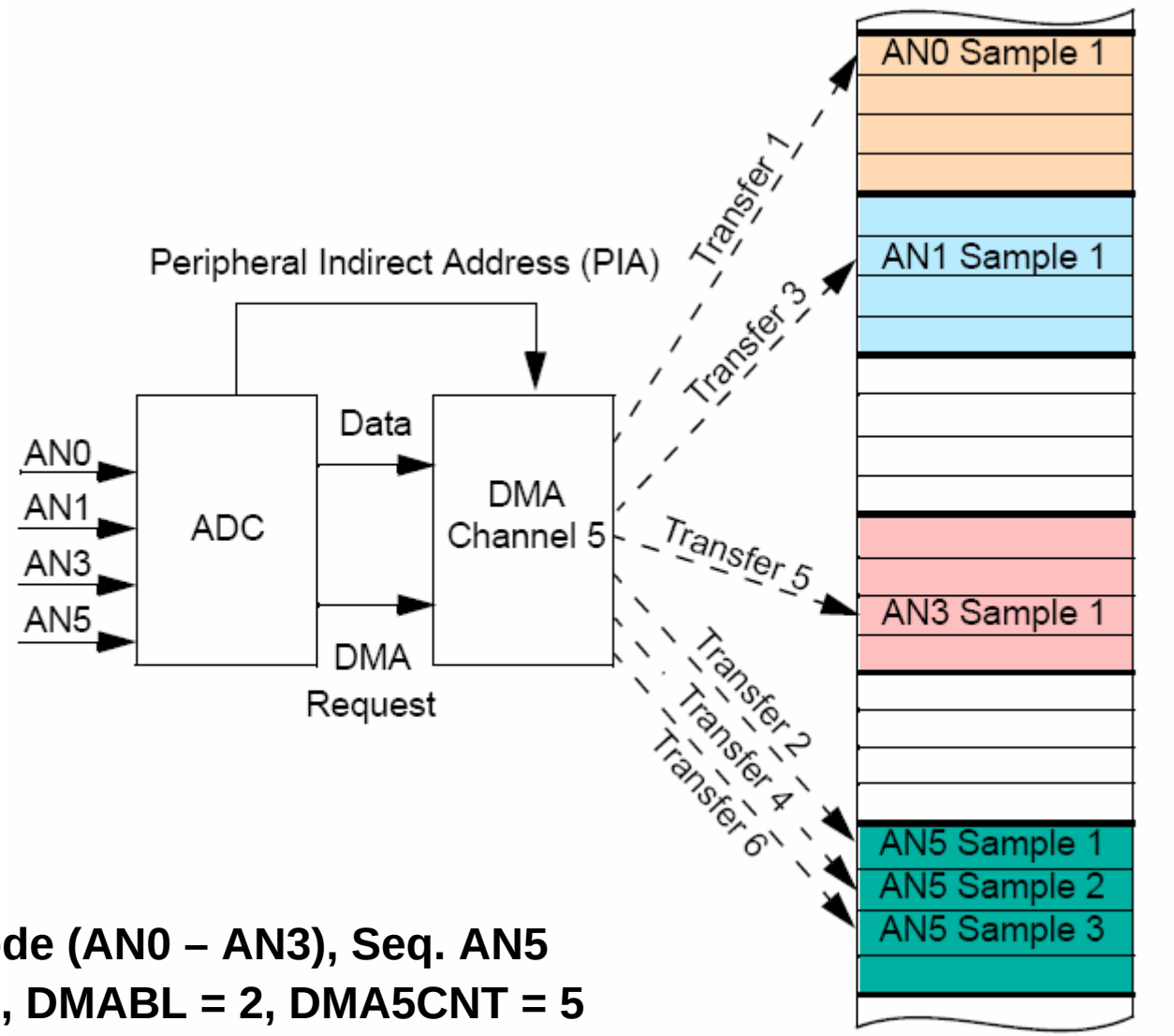
DMA Transfers – Peripheral Indirect (Gather Scatter)



- Scan mode (AN0 – AN3)
- SMPI = 3, DMABL = 2, DMA5CNT = 11
- DMA5STA -> last 4 bits must be 0's

© 2007 Microchip Technology Incorporated. All Rights Reserved.

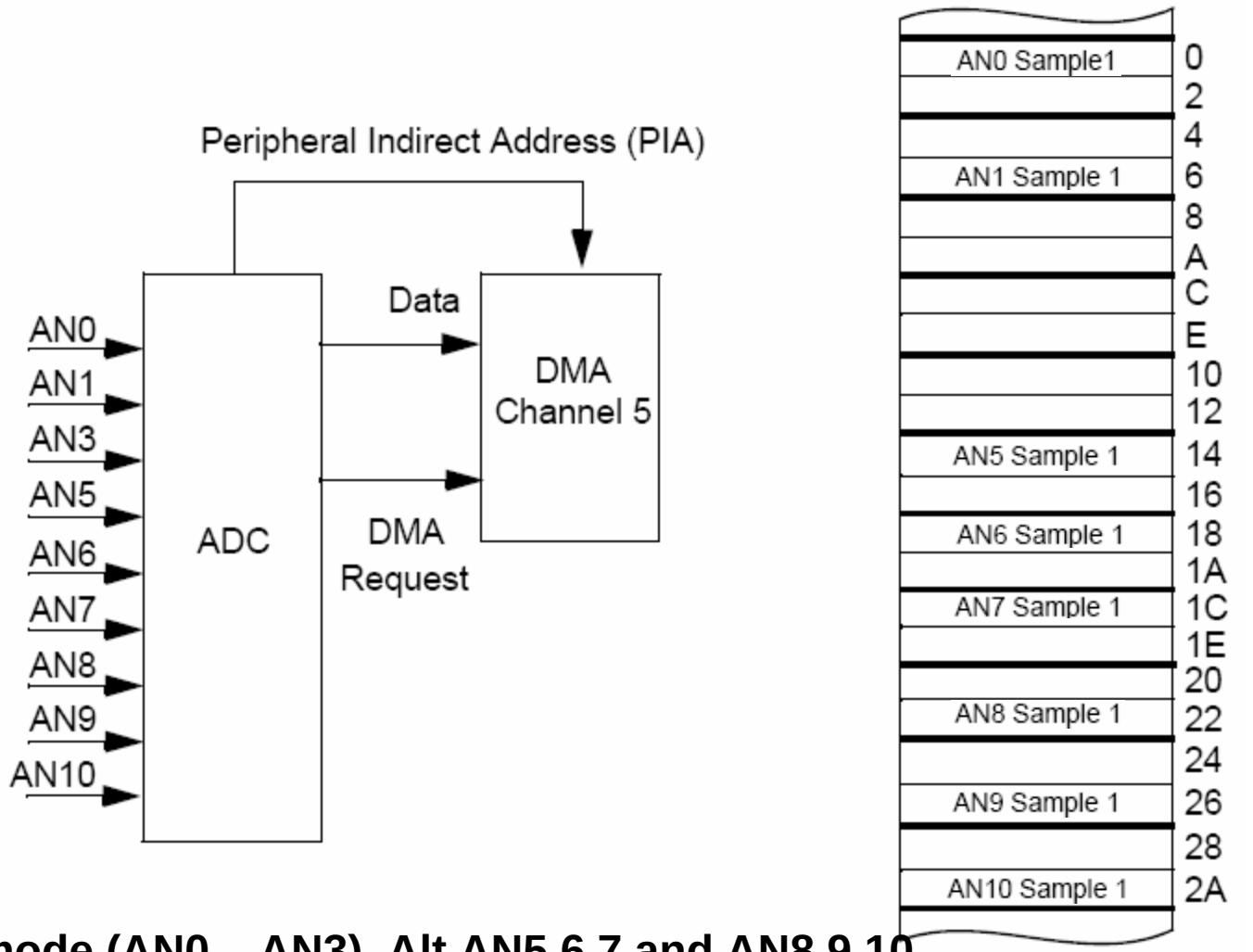
DMA Transfers – Peripheral Indirect (Gather Scatter)



- Scan mode (AN0 – AN3), Seq. AN5
- SMPI = 5, DMABL = 2, DMA5CNT = 5
- DMA5STA -> last 4 bits must be 0's

© 2007 Microchip Technology Incorporated. All Rights Reserved.

DMA Transfers – Peripheral Indirect (Gather Scatter)



- **Scan mode (AN0 – AN3), Alt AN5,6,7 and AN8,9,10**
- **SMPI = 10, DMABL = 1, DMA5CNT = 5**
- **DMA5STA -> last 4 bits must be 0's**

© 2007 Microchip Technology Incorporated. All Rights Reserved.

Lab 1

Sensor Application Example

© 2007 Microchip Technology Incorporated. All Rights Reserved.

Lab 2

Motor Control Application Example

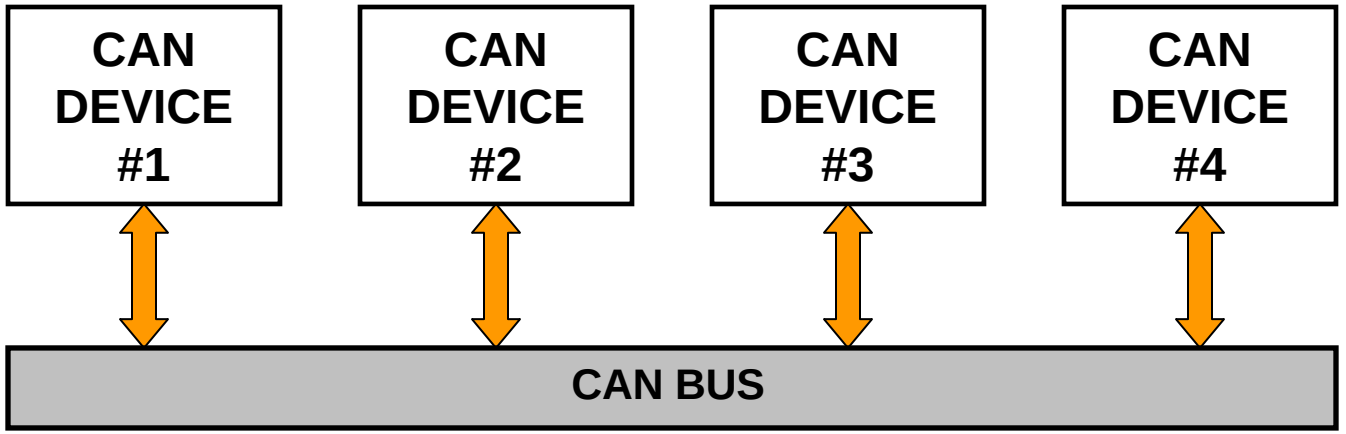
© 2007 Microchip Technology Incorporated. All Rights Reserved.

ECAN™ Module

© 2007 Microchip Technology Incorporated. All Rights Reserved.

CAN Protocol Overview

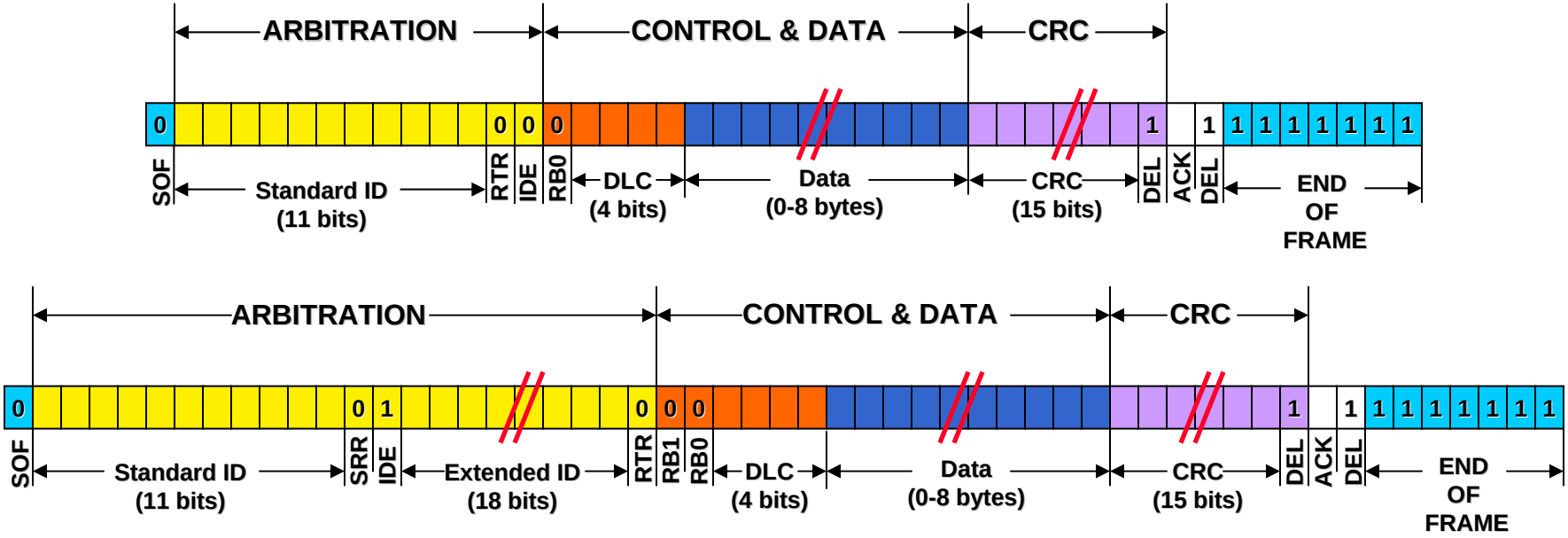
- The CAN bus is a serial communication protocol
- All nodes are connected together
- All nodes must use the same baud rate
- Each node can transmit or receive any message



© 2007 Microchip Technology Incorporated. All Rights Reserved.

CAN Data Frame

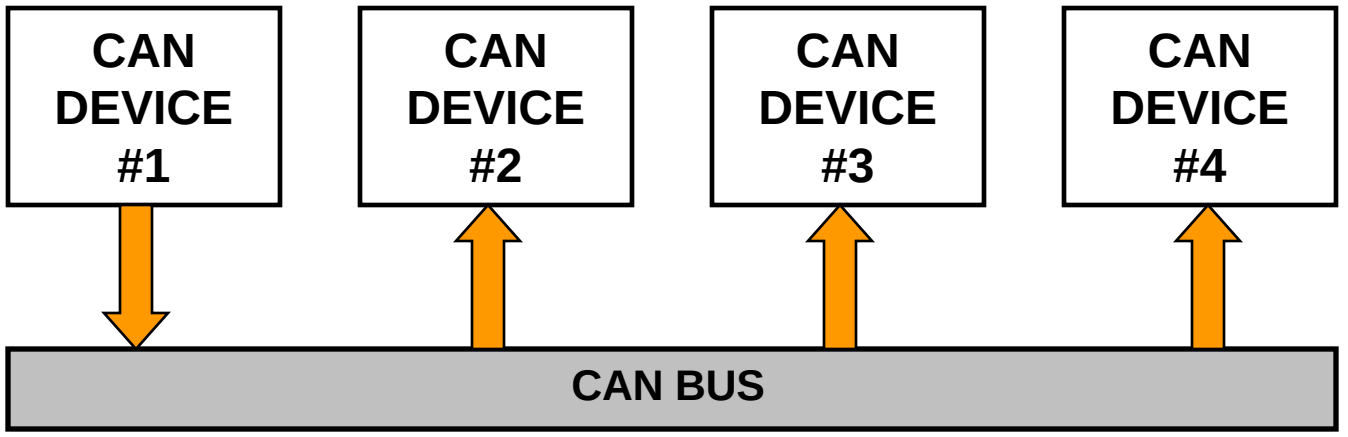
- **Identifier is at the start of the message**
  - Two formats exist for the identifier: Standard and Extended
- **Data contains control and data bytes of message**
- **Message has start, end, CRC and acknowledge overhead**



© 2007 Microchip Technology Incorporated. All Rights Reserved.

CAN Bus Operation

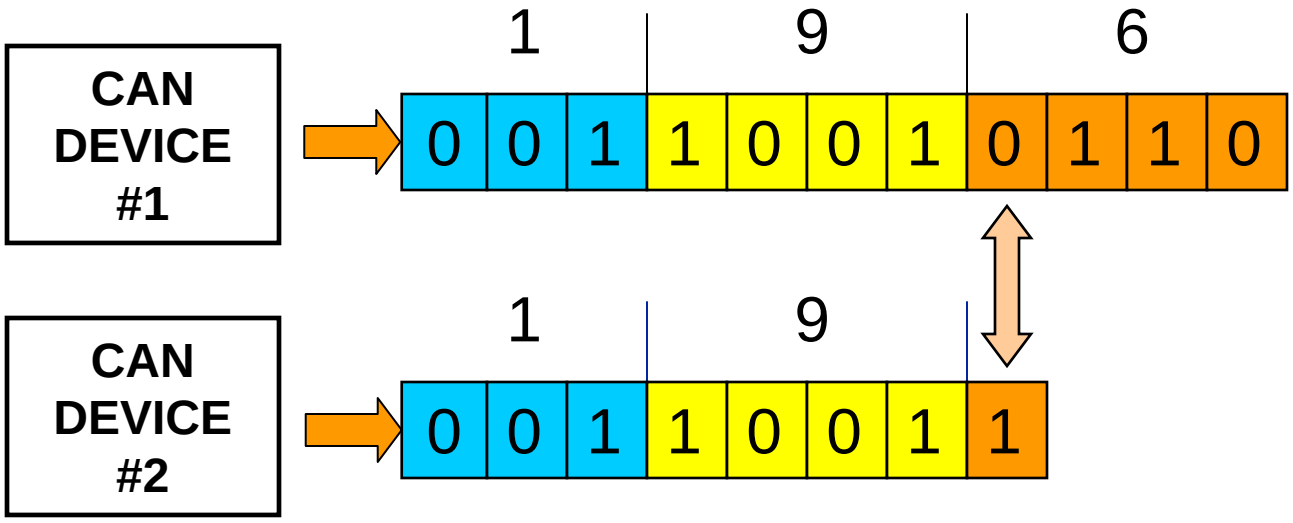
- Only one transmitter is allowed on the bus at a time
- The transmitter sends the message to all receivers
- All receivers will acknowledge reception of the message
- Receivers filter message identifiers



© 2007 Microchip Technology Incorporated. All Rights Reserved.

CAN Bus Arbitration

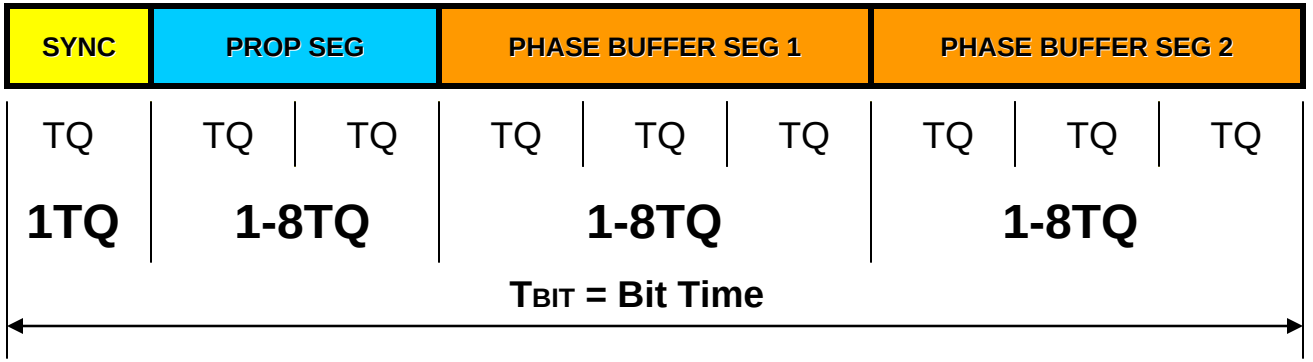
- Only one device can transmit at a time
- Multiple devices could start transmitting at same time
- Arbitration mechanism – device which transmits first zero wins



© 2007 Microchip Technology Incorporated. All Rights Reserved.

CAN Bus – Bit Details

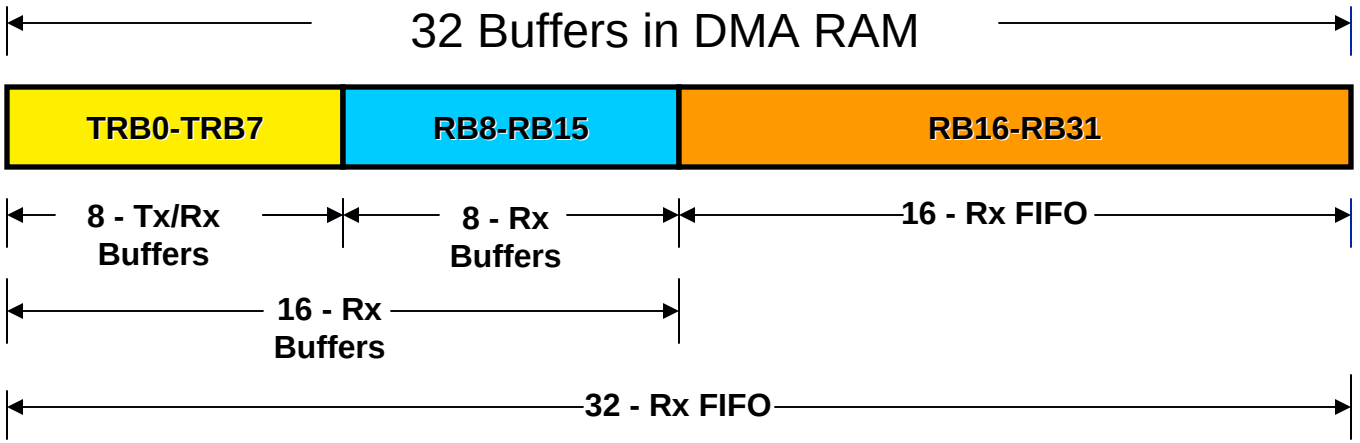
- A CAN message bit consists of 4 segments
- Each segment is composed of a Time Quanta (TQ)
- Bit time can range from 8 to 25 TQ



© 2007 Microchip Technology Incorporated. All Rights Reserved.

ECAN™ Buffers

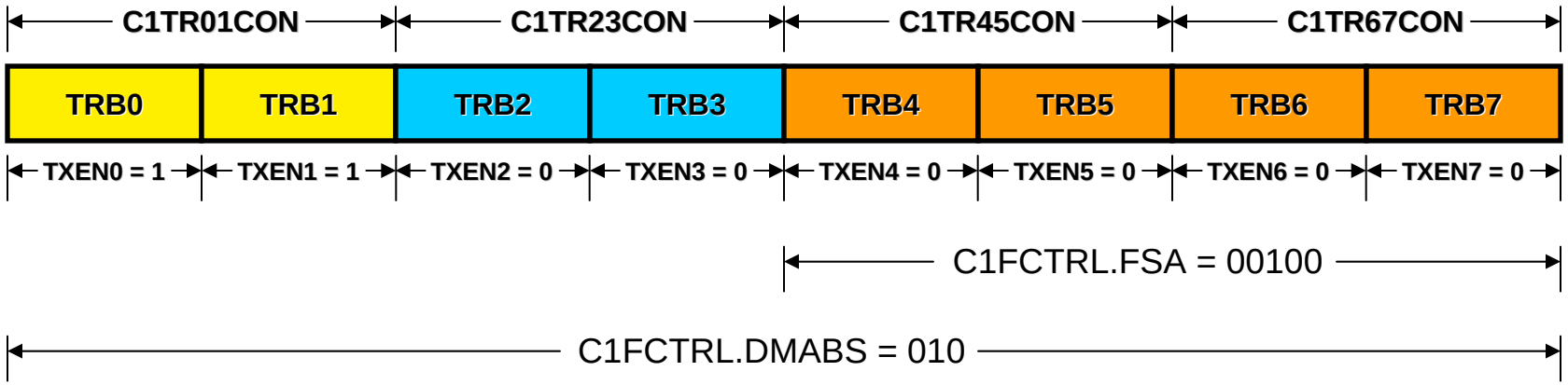
- **Up to 8 Transmit / Receive Buffers (TRB0-TRB7)**
- **Up to 24 additional receive only buffers(RB8-RB31)**
  - RB16-31 accessible as a FIFO
- **All 32 buffers in DMA RAM**
- **Each buffer is 16 bytes wide**



© 2007 Microchip Technology Incorporated. All Rights Reserved.

ECAN™ Buffers

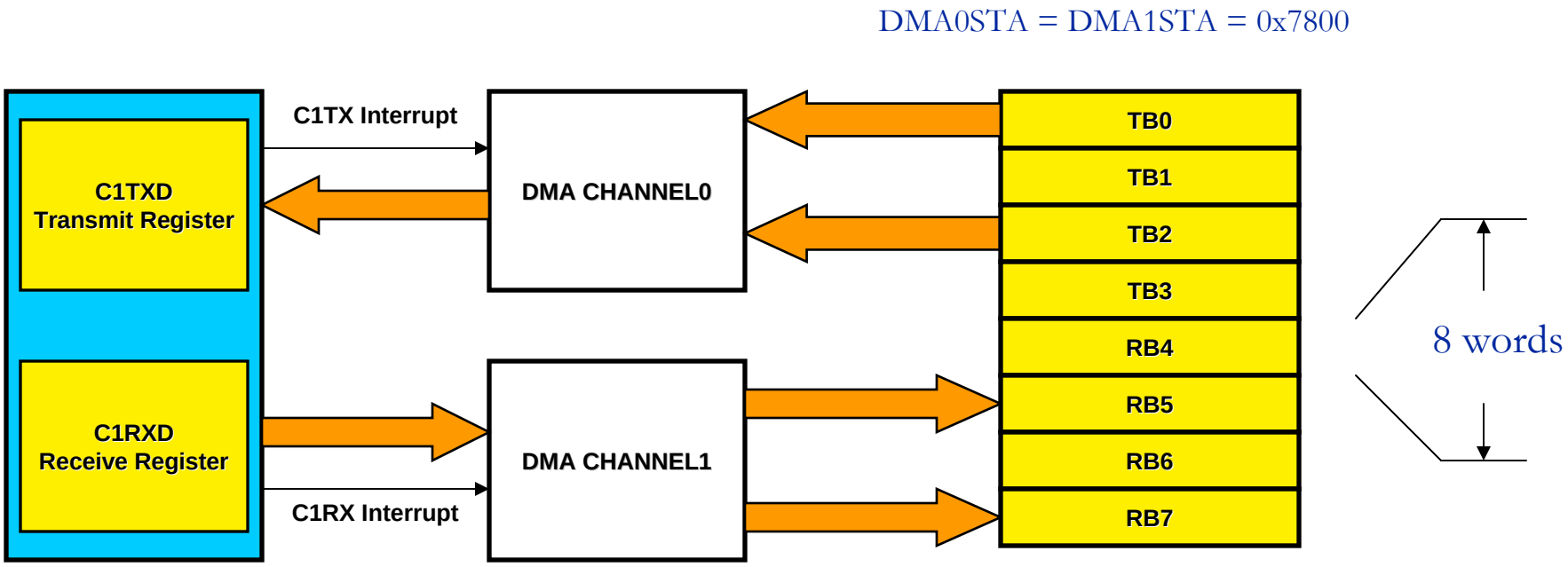
- TRB0 – TRB7 controlled via CiTRmnCON register
- DMABS bits (CiFCTRL register) decides total number of buffers
- FSA bits (CiFCTRL register) decides where FIFO starts



© 2007 Microchip Technology Incorporated. All Rights Reserved.

Message Buffers & DMA

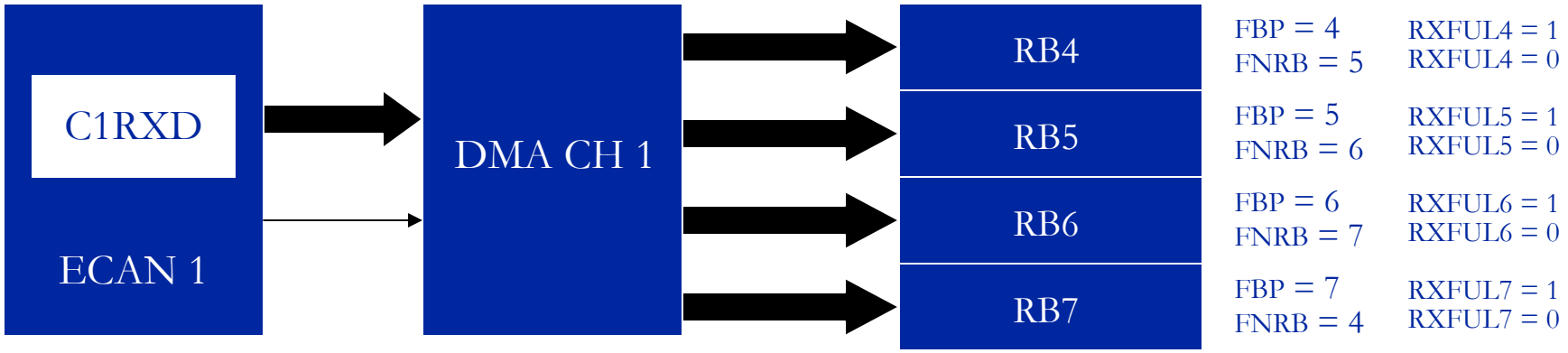
- ECAN™ module provides peripheral addressing to DMA
- DMASTAx value defines top of ECAN Buffer
- 2 DMA Channels per ECAN module typical
- Both channels point to the same DMA memory locations
- Buffer addresses offset from DMA memory start location



© 2007 Microchip Technology Incorporated. All Rights Reserved.

Message FIFO and DMA

- ECAN™ module maintains FIFO status
- FBP – current buffer pointer
- RXFULx bit indicates status of buffer in FIFO
- FNRB – next buffer to read



© 2007 Microchip Technology Incorporated. All Rights Reserved.

DMA Configuration

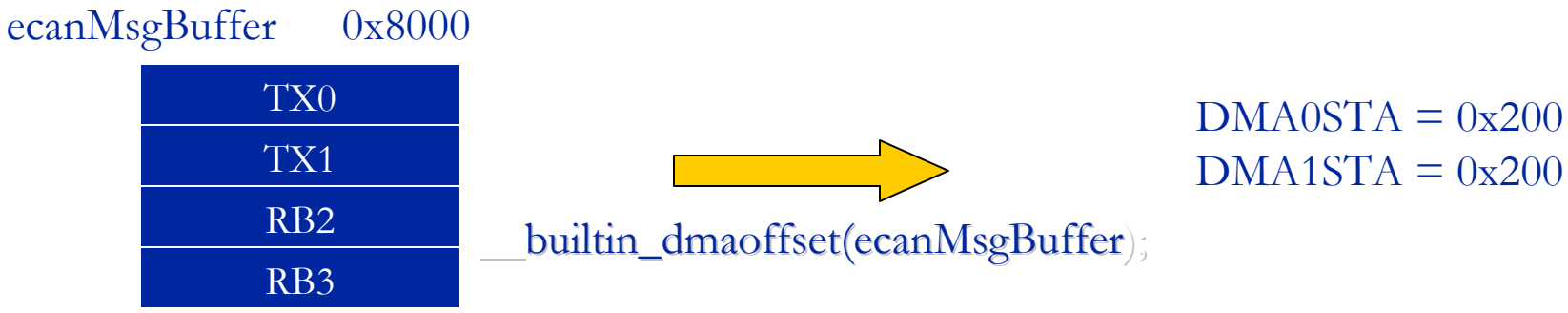
- **Peripheral Indirect Addressing**
- **One-shot Ping Mode disabled**
- **Word Transfer**
- **ECAN™ Module Transmit**
  - DMAxREQ = 0x46 (ECAN1) or 0x
  - DMAxPAD = 0x442(ECAN1) or
- **ECAN Module Receive**
  - DMAxREQ = 0x22 (ECAN1) or 0x
  - DMAxPAD = 0x440 (ECAN1) or
- **DMAxCNT = 7**

© 2007 Microchip Technology Incorporated. All Rights Reserved.

DMA Memory

- Use __attribute__((space(dma))) for allocation
- Use __builtin_dmaoffset() for obtaining offset

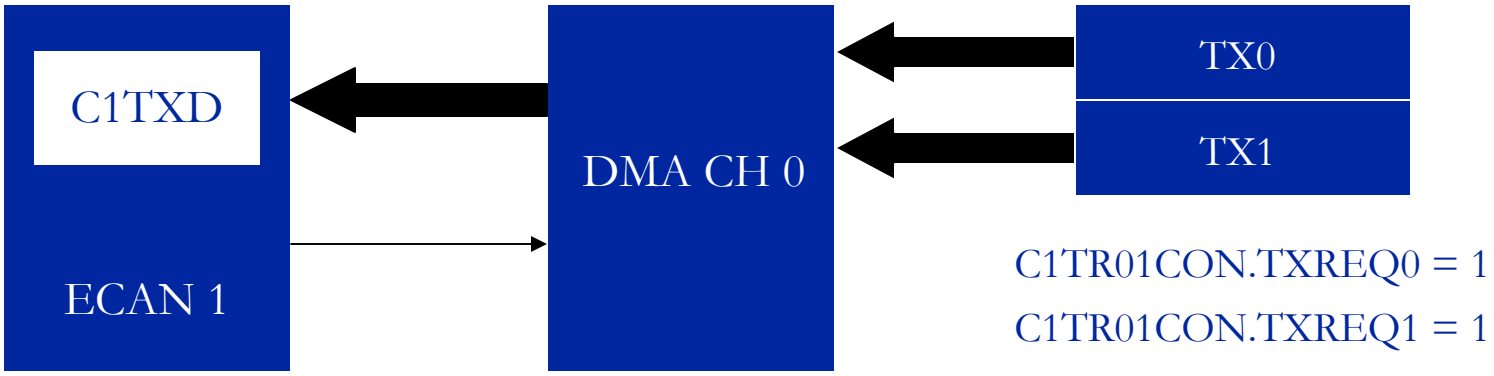
```
int ecanMsgBuffer [12][8] __attribute__((space(dma)));
DMA0STA = __builtin_dmaoffset(ecanMsgBuffer);
DMA1STA = __builtin_dmaoffset(ecanMsgBuffer);
```



© 2007 Microchip Technology Incorporated. All Rights Reserved.

ECAN™ Transmission

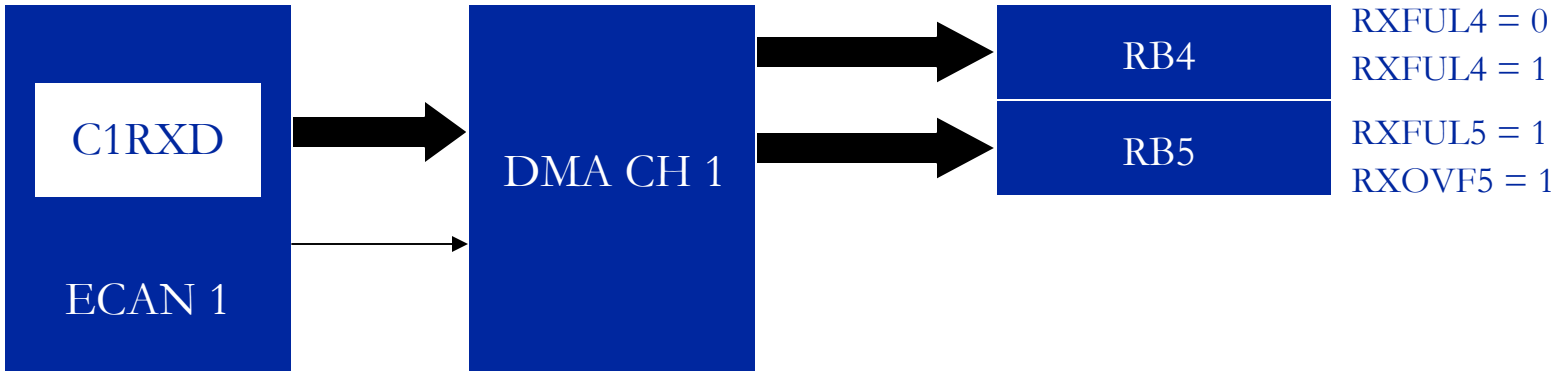
- Set the TXREQ bit in CiTRmnCON register
- ECAN module arbitrates priority
- Message is transmitted without user intervention
- TXREQ bit is cleared



© 2007 Microchip Technology Incorporated. All Rights Reserved.

ECAN™ Reception

- Received message matched against enabled filters
- For a successful match, message is stored into an assigned buffer
- RXFUL and RXOVF bits show buffer status



© 2007 Microchip Technology Incorporated. All Rights Reserved.

Message Filters

- Filters and Masks enable selective reception
- Filters registers: CiRXFnSID, CiRXFnEID, CiFEN and CiBUFPNT
- Mask registers: CiRXMnSID, CiRXMnEID and CiFMSKSEL

FILTER/MASK TRUTH TABLE			
Mask Bit n	Filter Bit n	Message Identifier Bit n	Accept or Reject bit n
0	X	X	Accept
1	0	0	Accept
1	0	1	Reject
1	1	0	Reject
1	1	1	Accept

© 2007 Microchip Technology Incorporated. All Rights Reserved.

ECAN™ Interrupts

- CiTX – ECAN Transmit Data Request
- CiRX – ECAN Receive Data Ready
- Ci – ECAN Event Interrupt

Interrupt occurs when a word in a message is transmitted

Interrupt occurs when a word in a message is received

Interrupt occurs when an entire message has been transmitted or received or for other ECAN module events

© 2007 Microchip Technology Incorporated. All Rights Reserved.

Message Buffer Format

	15 13	12 2	1	0
Word 0		SID <10:0>	SRR	IDE

	15 12	0
Word 1		EID <17:6>

	15 10	9	8	7 5	4	
Word 2	EID <5:0>	RTR	RB1		RB0	DLC <3:0>

Word 3	Data byte 1	Data byte 0
Word 4	Data byte 3	Data byte 2
Word 5	Data byte 5	Data byte 4
Word 6	Data byte 7	Data byte 6

	15 13	8	
Word 7		FILHIT <4:0>	

- TX0
- TX1
- RB2
- RB3
- RB4
- RB5
- RB6
- RB7

© 2007 Microchip Technology Incorporated. All Rights Reserved.

Steps to a Successful ECAN™ Module Experience

- **Configure ECAN baud rate**
- **Set WIN bit to enable Filter Register window**
  - Enable required number of filters
  - Set up Filter Identifier registers
  - Set up Filter Mask Identifier registers
  - Assigns Mask to filters
  - Set up Target buffers for filters

© 2007 Microchip Technology Incorporated. All Rights Reserved.

Steps to a Successful ECAN™ Module Experience

- **Set WIN bit to Zero to enable buffer window**
  - Set the total buffer size
  - Set the FIFO start area
  - Set up Transmit registers
- **Configure DMA**
  - Allocate DMA memory for buffers
  - Configure two channels (TX and RX)
  - Assign Peripheral address and interrupt sources
  - Enable the channels

© 2007 Microchip Technology Incorporated. All Rights Reserved.

Steps to a Successful ECAN™ Module Experience

- Enable ECAN event interrupt to track error events if needed
- Enable DMA interrupts
- Set ECAN mode to operational
- Set TXREQ bits to transmit a buffer
- Check RXFUL flags in ECAN Receive DMA Channel interrupt

© 2007 Microchip Technology Incorporated. All Rights Reserved.

Lab 3

ECAN™ Communication

© 2007 Microchip Technology Incorporated. All Rights Reserved.

University of Microchip UofM MASTERs 2007

Summary

- High Performance ADC Features
- Flexible ADC Input & Conversion Options
- How the ECAN™ Module Works
- Benefits of using DMA with ADC and ECAN

© 2007 Microchip Technology Incorporated. All Rights Reserved.

Thank You !

© 2007 Microchip Technology Incorporated. All Rights Reserved.

UNIVERSITY OF MICROCHIP UOFM MASTERS 2007

Trademarks

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KeeLoq, KeeLoq logo, microID, MPLAB, PIC, PICmicro, PICSTART, PRO MATE, rfPIC and SmartShunt are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AmpLab, FilterLab, Linear Active Thermistor, Migratable Memory, MXDEV, MXLAB, SEEVAL, SmartSensor and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, ECAN, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, PICkit, PICDEM, PICDEM.net, PICLAB, PICtail, PowerCal, PowerInfo, PowerMate, PowerTool, REAL ICE, rfLAB, Select Mode, Smart Serial, SmartTel, Total Endurance, UNI/O, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2007 Microchip Technology Incorporated. All Rights Reserved.