Increasing Data Throughput Using the New C2000 DMA

Overview of the TMS320F2833x DMA

Agenda

- F2833x Block Diagram/Overview
- Channel Capabilities
- State Machine
- Address Control
- Other Features/Register Descriptions
- Examples

Agenda

- F2833x Block Diagram/Overview
- Channel Capabilities
- State Machine
- Address Control
- Other Features/Register Descriptions
- Examples

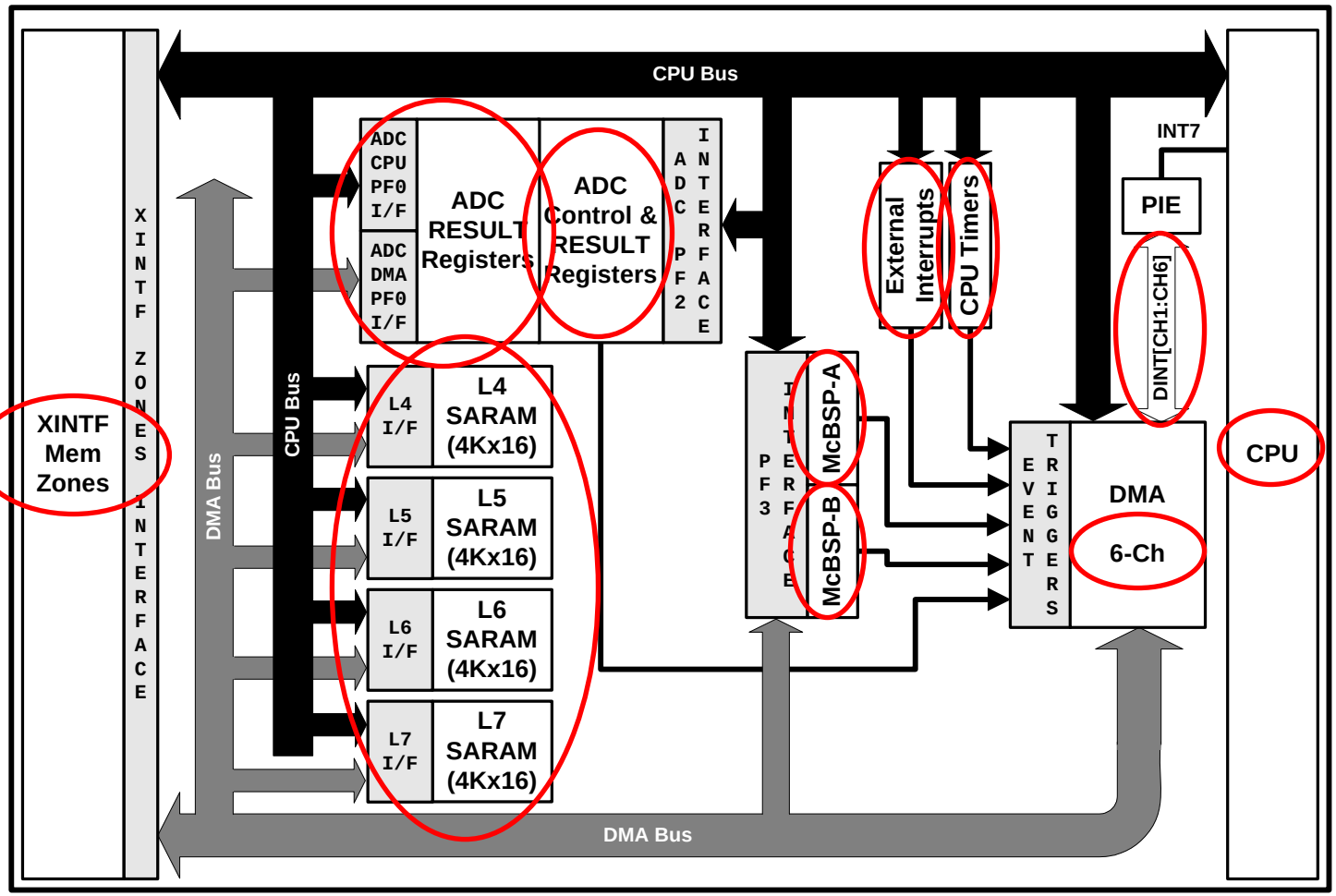
Why DMA for Control?

- Reduce ADC overhead, e.g., bar-code readers, data acquisition
- Processing of external data, e.g. FFT on external data
- Binning ADC data
- Data logging and communication via McBSP
- Transposing of matrices, e.g., power system monitoring
- Any block data transfers

F2833x DMA Overview

- Event driven state machine
- 6 channels can be configured to accept any DMA source
- Transfers split up into smaller bursts
- Flexible automated address manipulation
- 16 bit or 32 bit word transfers
- Fully configurable

F2833x DMA Block Diagram



6 channels w/ interrupt capabilities

74 possible event triggers

5 data location possibilities

Agenda

- ✓ F2833x Block Diagram/Overview
- ❑ **Channel Capabilities**
- ❑ State Machine
- ❑ Address Control
- ❑ Other Features/Register Descriptions
- ❑ Examples

Channel Input Sources

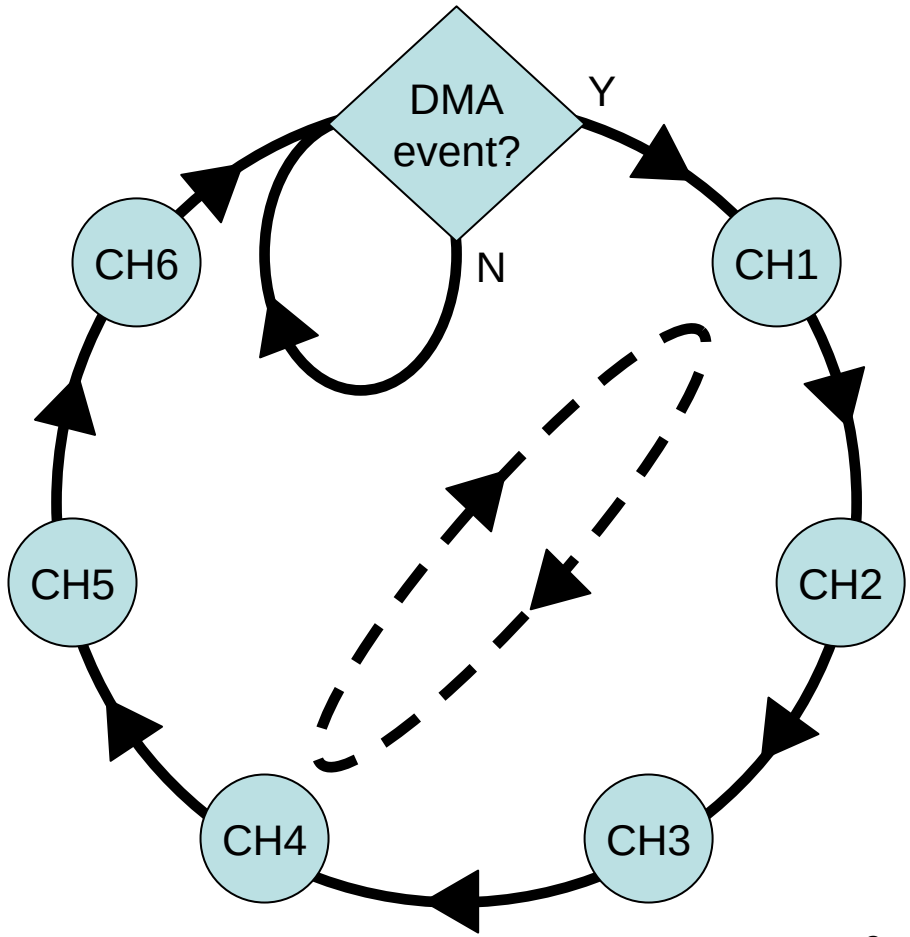
MODE.CHx[PERINTSEL]

Value	Interrupt	Peripheral
0	none	No Peripheral Connection
1	SEQ1INT	ADC
2	SEQ2INT	
3	XINT1	External Interrupts
4	XINT2	
5	XINT3	
6	XINT4	
7	XINT5	
8	XINT6	
9	XINT7	
10	XINT13	
11	TINT0	CPU Timers
12	TINT1	
13	TINT2	
14	MXEVTA	McBSP-A
15	MREVTA	
16	MXEVTB	McBSP-B
17	MREVTB	
31:18	spare	No peripheral connection.

Software (anytime)	1
ADC Sequencer 1 & 2	2
3 XINT (GPIO31:0 pins)	32
5 XINT (GPIO63:32 pins)	32
Timer0, Timer1, Timer 2	3
McBSP-A Tx & Rx	2
McBSP-B Tx & Rx	2
Total	74

Channel Priorities

- Two modes:
  - Round robin
  - Channel 1 priority: same as round robin except CH1 can interrupt DMA state machine
- Channel 1 priority mode uses:
  - Primarily for ADC
  - Can be used by any peripheral configured to trigger CH1

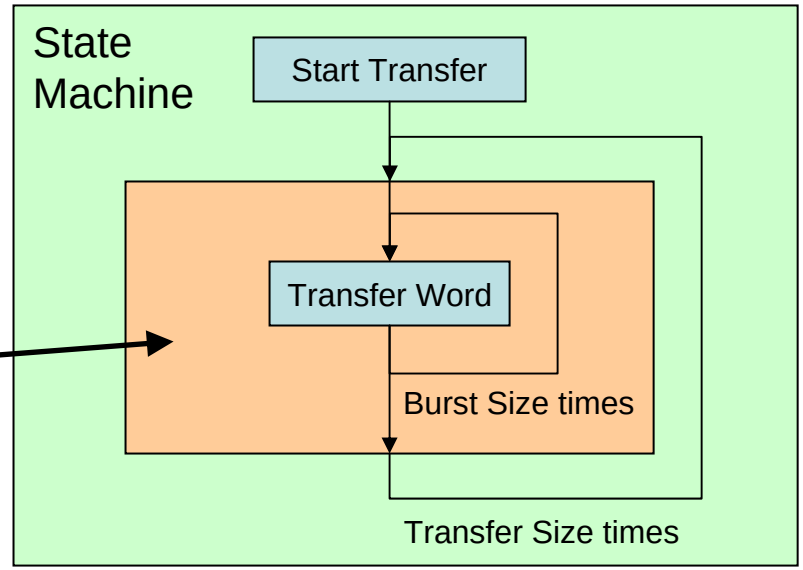


Agenda

- ✓ F2833x Block Diagram/Overview
- ✓ Channel Capabilities
- ❑ **State Machine**
- ❑ Address Control
- ❑ Other Features/Register Descriptions
- ❑ Examples

State Machine Definitions

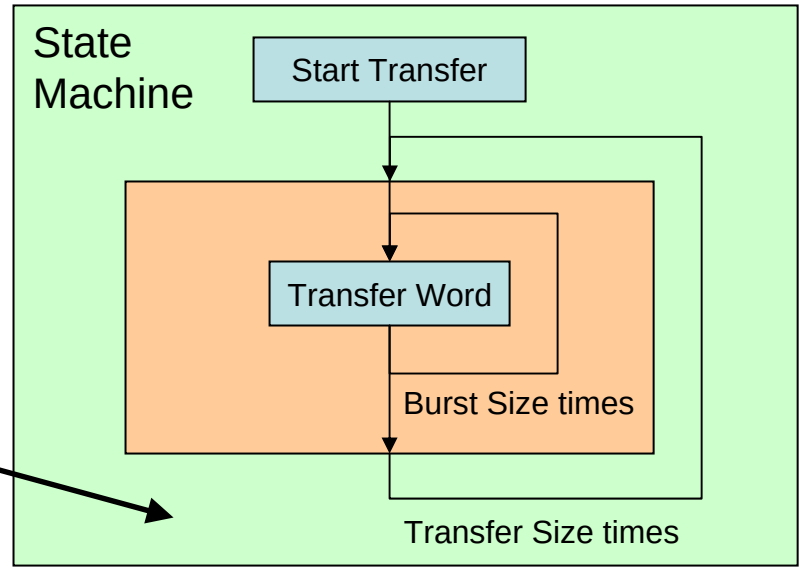
- Burst – smallest amount of data moved at a time
- Burst Size – words/burst
- BURST_SIZE – Burst Size - 1
- Burst Step – added to address pointer inside burst loop



Note: Step registers are all 16 bit signed values
Size registers are all N-1 values

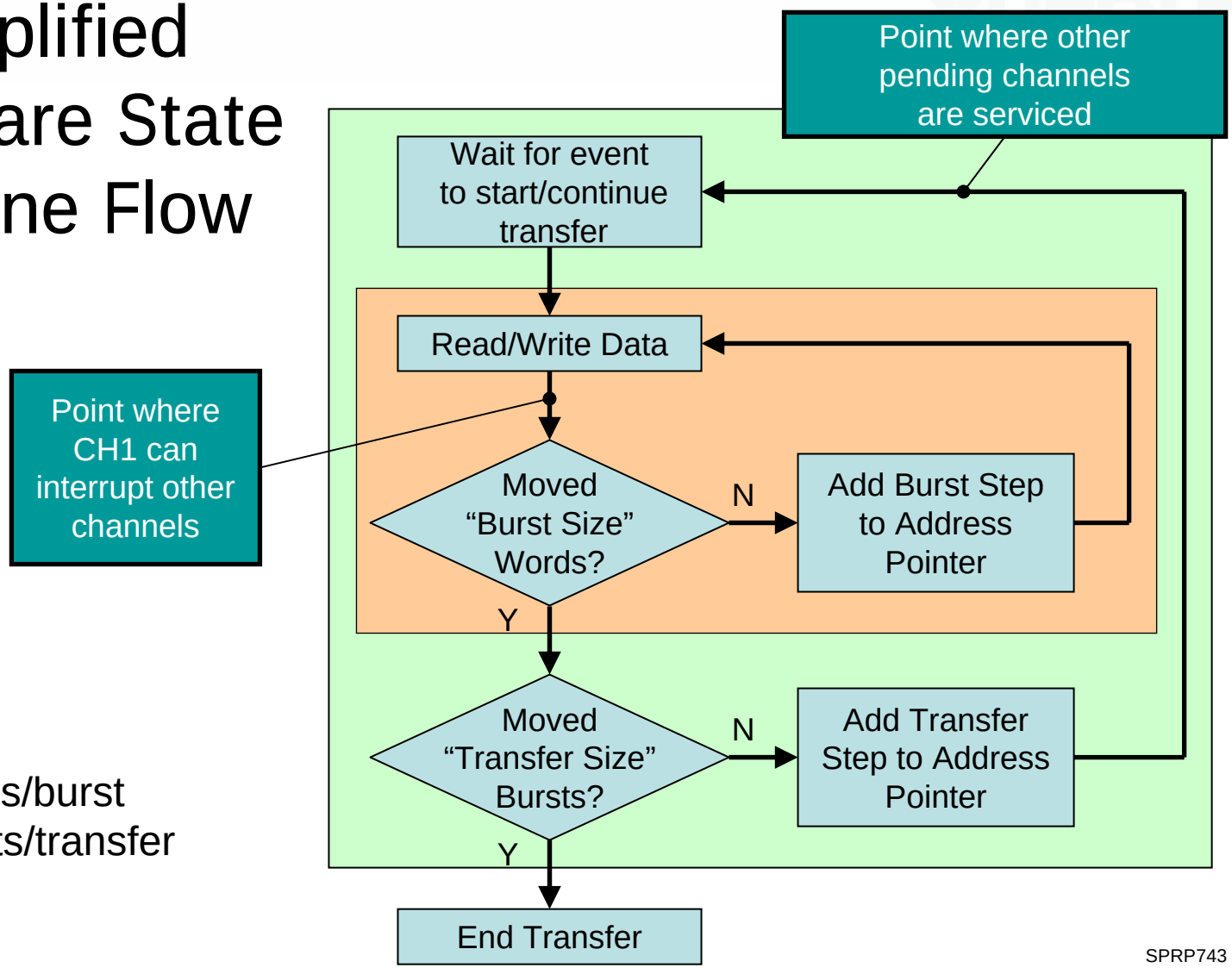
State Machine Definitions

- Transfer – period between CPU interrupts
- Transfer Size – bursts/transfer
- TRANSFER_SIZE – Transfer Size - 1
- Transfer Step – added to address pointer inside transfer loop



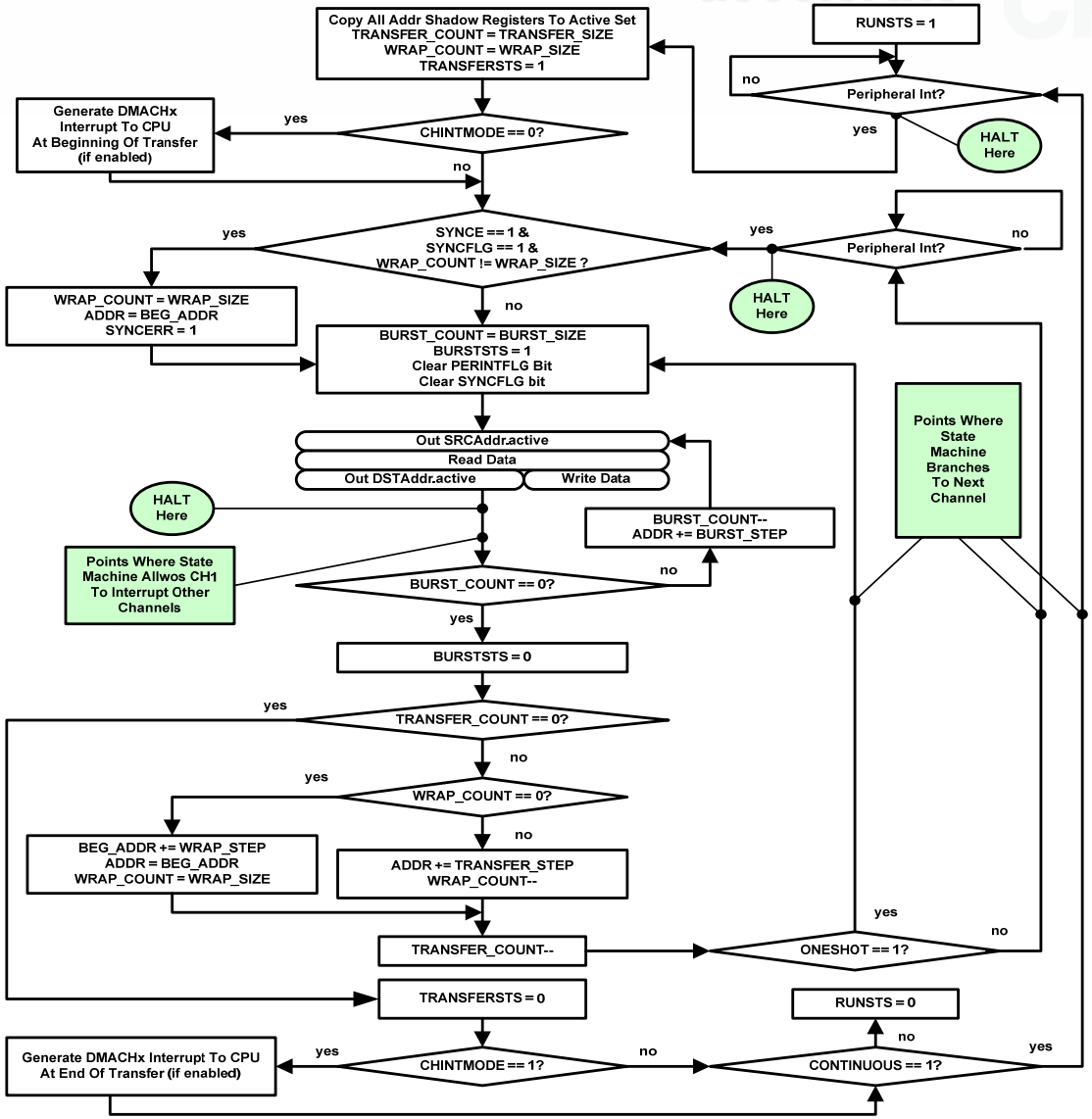
Note: Step registers are all 16 bit signed values
Size registers are all N-1 values

Simplified Hardware State Machine Flow



Example:

- 3 words/burst
- 2 bursts/transfer



DMA State Machine Throughput

- 4 cycles/word (5 for McBSP reads)
- 1 cycle delay to start each burst
- 1 cycle delay returning from CH1 high priority interrupt
- 32-bit transfer doubles throughput (except McBSP)

Example: 128 16-bit words from ADC to RAM

8 bursts * [(4 cycles/word * 16 words/burst) + 1] = **520 cycles**

Example: 64 32-bit words from ADC to RAM

8 bursts * [(4 cycles/word * *8 words/burst*) + 1] = **264 cycles**

Agenda

- ✓ F2833x Block Diagram/Overview
- ✓ Channel Capabilities
- ✓ State Machine
- ❑ **Address Control**
- ❑ Other Features/Register Descriptions
- ❑ Examples

Address Control

- Separate source and destination pointers with separate control
- Buffered pointers to allow ping-pong
- Signed number added after each word transferred
- Signed number added after each burst completed
- Wrap with offset option available

Address Control

BURST_SIZE*	0x0001	2 words/burst
TRANSFER_SIZE*	0x0001	2 bursts/transfer

* Size registers are N-1

Source Registers

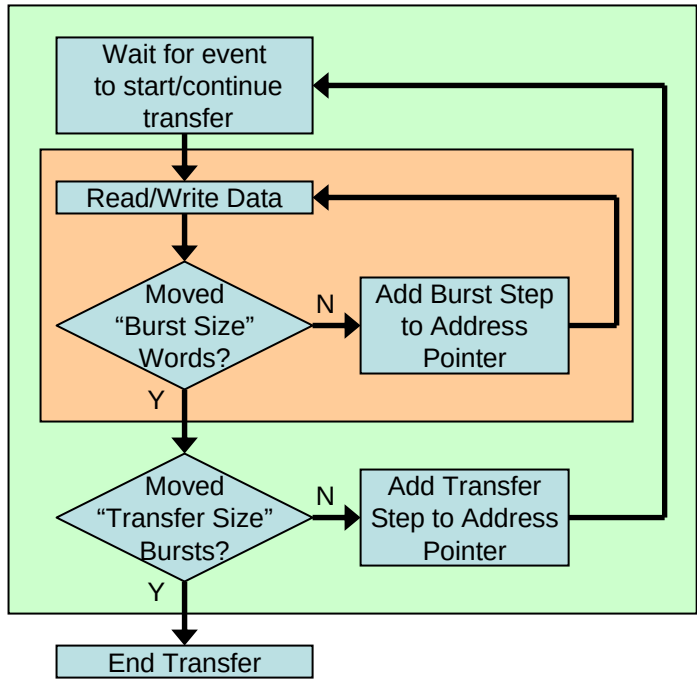
ADDR.active	0x00000000
ADDR.shadow	0x0000F000
BURST_STEP	0x0001
TRANSFER_STEP	0x0001

Addr	Value
0xF000	0x1111
0xF001	0x2222
0xF002	0x3333
0xF003	0x4444

Destination Registers

ADDR.active	0x00000000
ADDR.shadow	0x0000C000
BURST_STEP	0x0001
TRANSFER_STEP	0x0001

Addr	Value
0xC000	0x0000
0xC001	0x0000
0xC002	0x0000
0xC003	0x0000

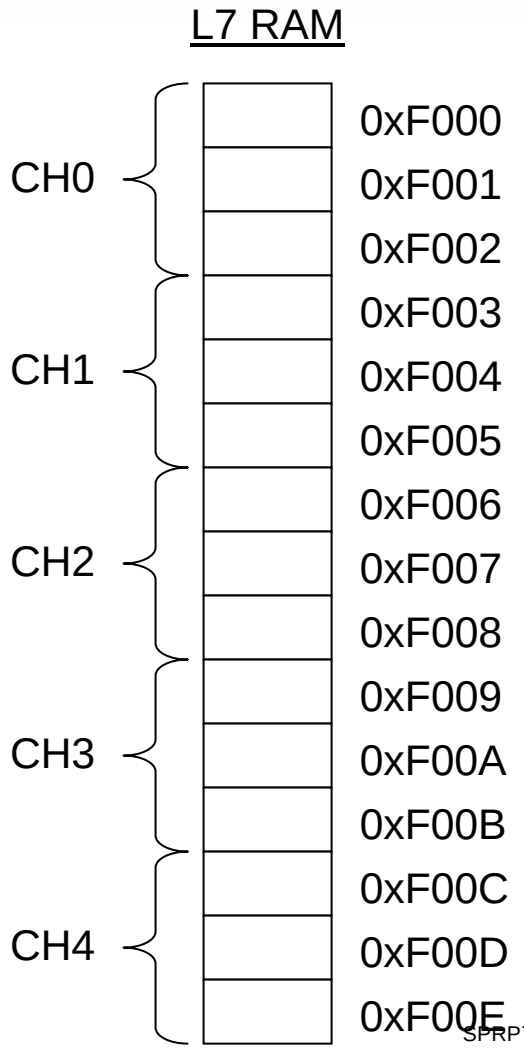


Address Control

ADC Result Registers

2nd Conversion Sequence

0x0B00	CH0
0x0B01	CH1
0x0B02	CH2
0x0B03	CH3
0x0B04	CH4



Objective: Bin ADC data into L7 RAM.
Interrupt CPU after 3 samples of 5 channels.

Address Control

BURST_SIZE*	0x0004	5 words/burst
TRANSFER_SIZE*	0x0002	3 bursts/transfer

* Size registers are N-1

Source Registers

ADDR.shadow	0x00000B00	
BURST_STEP	0x0001	
TRANSFER_STEP	0xFFFC	(-4)

Destination Registers

ADDR.shadow	0x0000F000	
BURST_STEP	0x0003	
TRANSFER_STEP	0xFFF5	(-11)

ADC Result Registers

0x0B00	CH0	CH0	CH0
0x0B01	CH1	CH1	CH1
0x0B02	CH2	CH2	CH2
0x0B03	CH3	CH3	CH3
0x0B04	CH4	CH4	CH4

time →

L7 RAM

0xF000	CH0
0xF001	CH0
0xF002	CH0
0xF003	CH1
0xF004	CH1
0xF005	CH1
0xF006	CH2
0xF007	CH2
0xF008	CH2
0xF009	CH3
0xF00A	CH3
0xF00B	CH3
0xF00C	CH4
0xF00D	CH4
0xF00E	CH4

Wrap Function

- Reloads address pointer after specified number of bursts
- Allows a cumulative signed offset to be added each wrap
- Provides another resource to manipulate the address pointers

Wrap Function Definitions

- Wrap Size – bursts/wrap
- WRAP_SIZE – Wrap Size – 1
- BEG_ADDR – Wrap beginning address
- Wrap Step – added to BEG_ADDR before wrapping

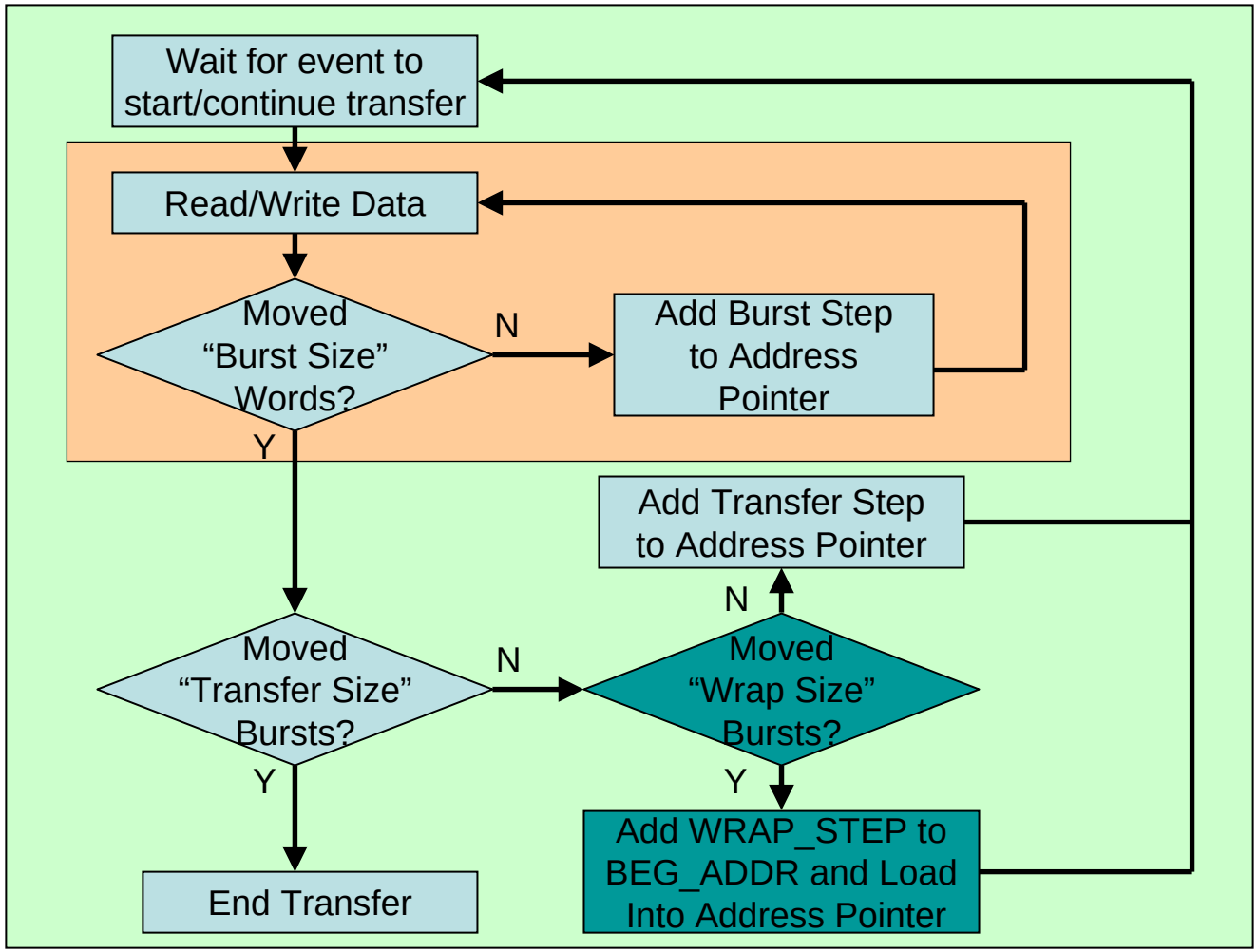
Wrap procedure:

Part 1: BEG_ADDR.active += WRAP_STEP

Part 2: ADDR.active = BEG_ADDR.active

Note: Step registers are all 16 bit signed values
Size registers are all N-1 values

Less Simplified State Machine Flow



Wrap Function

BURST_SIZE*	0x0004	5 words/burst
TRANSFER_SIZE*	0x0002	3 bursts/transfer

* Size registers are N-1

Source Registers

ADDR.shadow	0x00000B00	
BEG_ADDR.shadow	Don't Care	
BURST_STEP	0x0001	
TRANSFER_STEP	0xFFFC	(-4)
WRAP_STEP	Don't Care	
WRAP_SIZE*	0xFFFF	No wrap

Destination Registers

ADDR.shadow	0x0000F000	
BEG_ADDR.shadow	0x0000F000	
BURST_STEP	0x0003	
TRANSFER_STEP	Don't Care	
WRAP_STEP	0x0001	
WRAP_SIZE*	0x0000	1 burst/wrap

ADC Result Registers

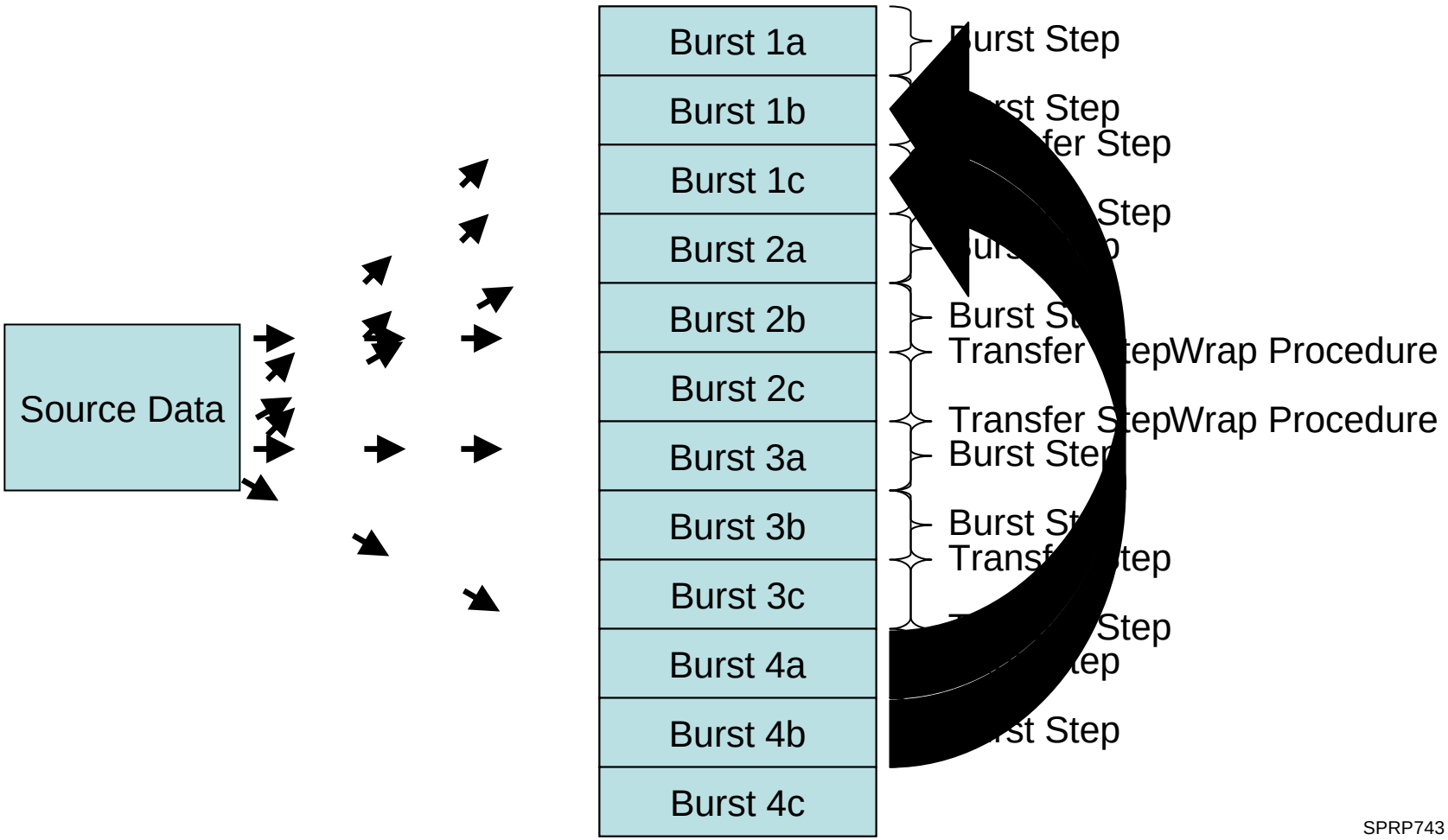
0x0B00	CH0	CH0	CH0
0x0B01	CH1	CH1	CH1
0x0B02	CH2	CH2	CH2
0x0B03	CH3	CH3	CH3
0x0B04	CH4	CH4	CH4

time →

L7 RAM

0xF000	CH0
0xF001	CH0
0xF002	CH0
0xF003	CH1
0xF004	CH1
0xF005	CH1
0xF006	CH2
0xF007	CH2
0xF008	CH2
0xF009	CH3
0xF00A	CH3
0xF00B	CH3
0xF00C	CH4
0xF00D	CH4
0xF00E	CH4

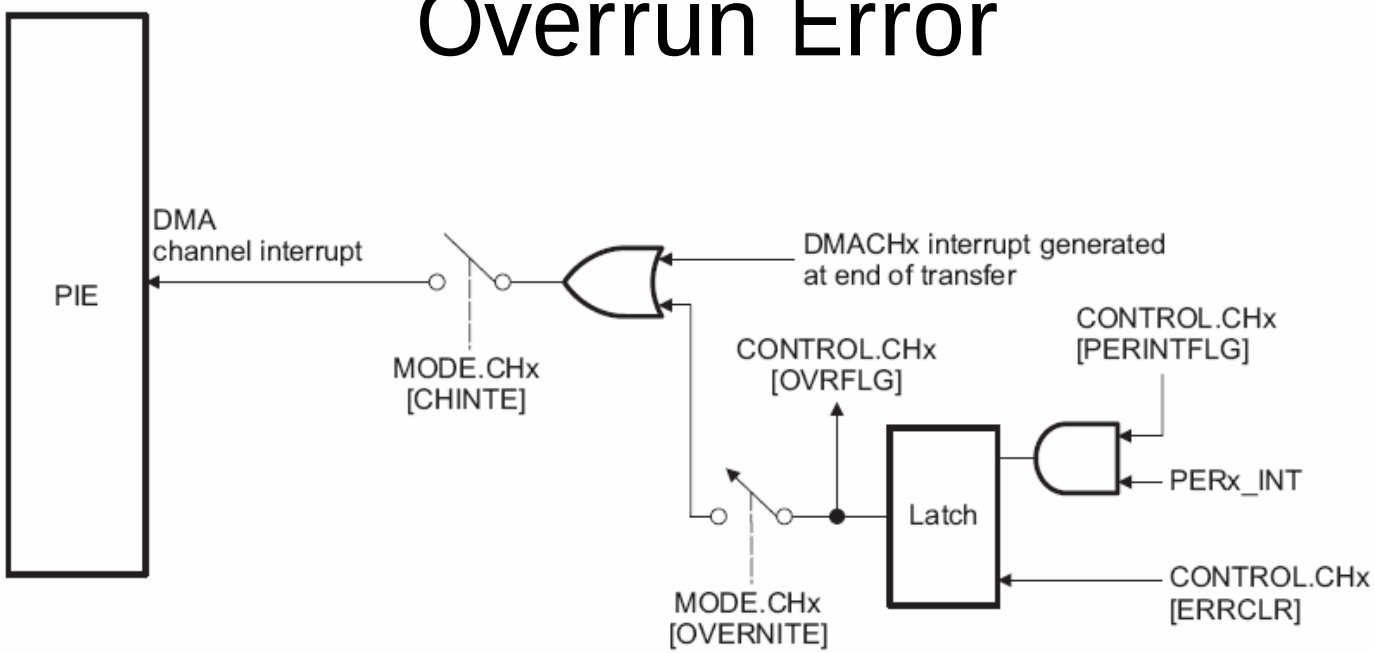
Multiple Buffers



Agenda

- ✓ F2833x Block Diagram/Overview
- ✓ Channel Capabilities
- ✓ State Machine
- ✓ Address Control
- ❑ **Other Features/Register Descriptions**
- ❑ Examples

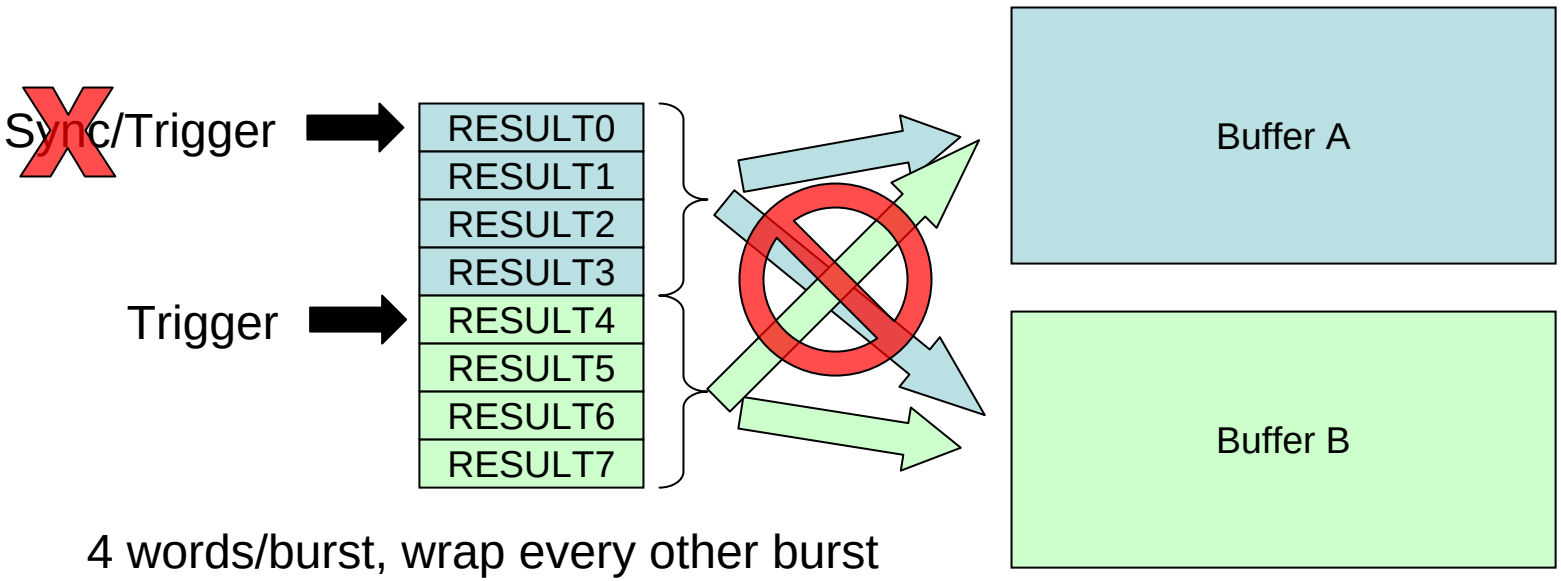
Overrun Error



- Warning to CPU that a DMA transfer was requested but not performed
- Triggered if a DMA transfer is pending and an additional request is received

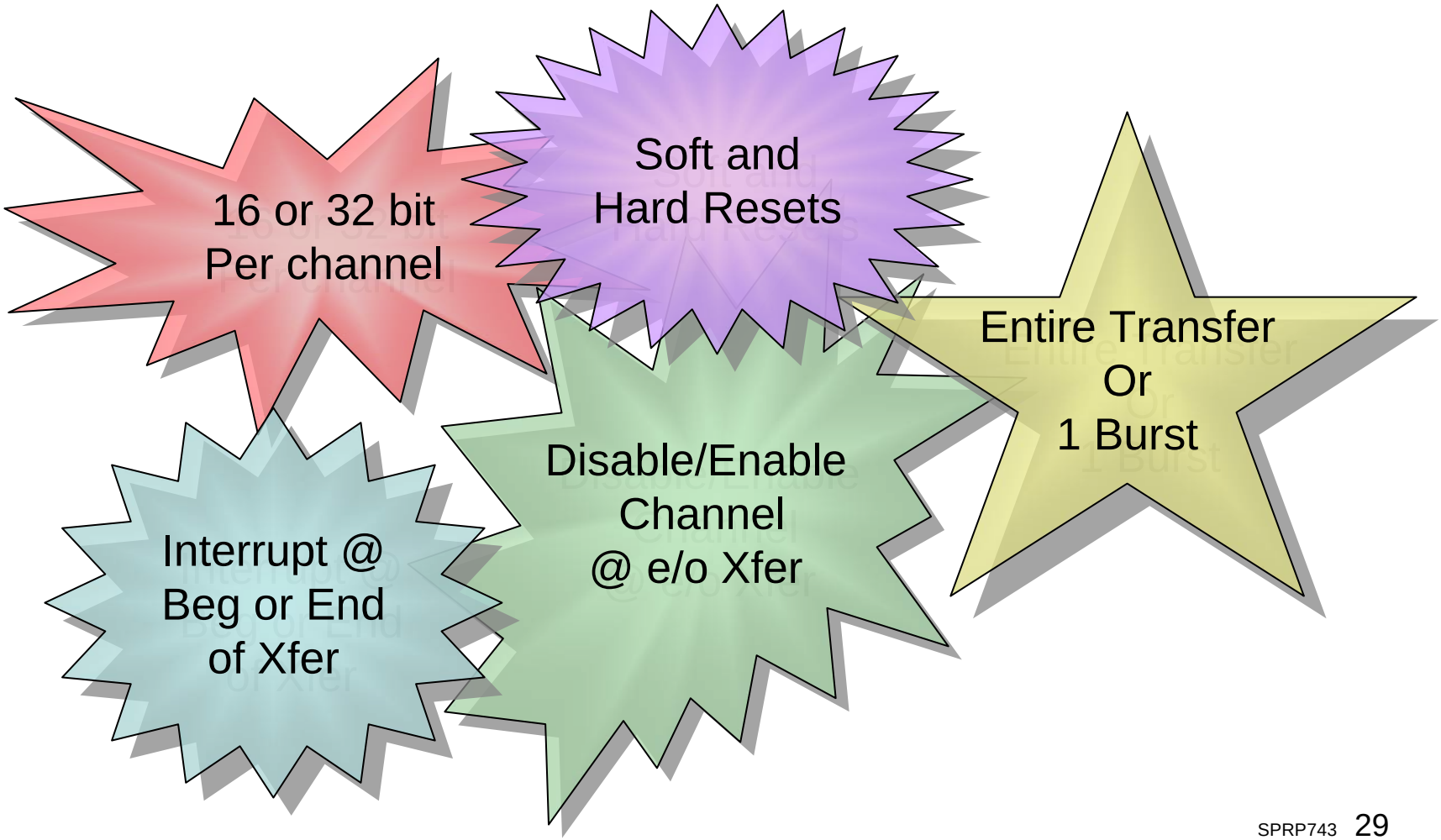
ADC Sync Feature

- Used when the ADC is operating in continuous conversion mode with the sequencer override bit set
- Automatically syncs two buffers to the conversions
- Generates error if sync does not correspond to wrap



4 words/burst, wrap every other burst

Other Features



DMA Register Summary

- EALLOW protected
- CH2 – CH6 registers imitate CH1 registers

Address	Acronym	Description
DMA Control, Mode and Status Registers		
0x1000	DMACTRL	DMA Control Register
0x1001	DEBUGCTRL	Debug Control Register
0x1002	REVISION	Peripheral Revision Register
0x1003	Reserved	Reserved
0x1004	PRIORITYCTRL1	Priority Control Register 1
0x1005	Reserved	Reserved
0x1006	PRIORITYSTAT	Priority Status Register
0x1007 0x101F	Reserved	Reserved
DMA Channel 1 Registers		
0x1020	MODE	Mode Register
0x1021	CONTROL	Control Register
0x1022	BURST_SIZE	Burst Size Register
0x1023	BURST_COUNT	Burst Count Register
0x1024	SRC_BURST_STEP	Source Burst Step Size Register
0x1025	DST_BURST_STEP	Destination Burst Step Size Register
0x1026	TRANSFER_SIZE	Transfer Size Register
0x1027	TRANSFER_COUNT	Transfer Count Register
0x1028	SRC_TRANSFER_STEP	Source Transfer Step Size Register
0x1029	DST_TRANSFER_STEP	Destination Transfer Step Size Register
0x102A	SRC_WRAP_SIZE	Source Wrap Size Register
0x102B	SRC_WRAP_COUNT	Source Wrap Count Register
0x102C	SRC_WRAP_STEP	Source Wrap Step Size Register
0x102D	DST_WRAP_SIZE	Destination Wrap Size Register
0x102E	DST_WRAP_COUNT	Destination Wrap Count Register
0x102F	DST_WRAP_STEP	Destination Wrap Step Size Register
0x1030	SRC_BEG_ADDR_SHADOW	Shadow Source Begin and Current Address Pointer Registers
0x1032	SRC_ADDR_SHADOW	
0x1034	SRC_BEG_ADDR	Active Source Begin and Current Address Pointer Registers
0x1036	SRC_ADDR	
0x1038	DST_BEG_ADDR_SHADOW	Shadow Destination Begin and Current Address Pointer Registers
0x103A	DST_ADDR_SHADOW	
0x103C	DST_BEG_ADDR	Active Destination Begin and Current Address Pointer Registers
0x103E	DST_ADDR	
0x103F	Reserved	Reserved

MODE Register

15	14	13	12	11	10	9	8
CHINTE	DATASIZE	SYNCSEL	SYNCE	CONTINUOUS	ONESHOT	CHINTMODE	PERINTE
R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0

7	6	5	4	0
OVRINTE	Reserved		PERINTSEL	
R/W - 0	R - 0		R/W - 0	

CHINTE	Channel interrupt enable to PIE module
DATASIZE	Chooses 16 or 32 bit transfers
SYNCSEL	Select whether sync is for source or destination registers
SYNCE	Sync function enable
CONTINUOUS	Determines whether or not channel is disabled at end of transfer
ONESHOT	Determines whether or not a transfer stops between bursts
CHINTMODE	Channel interrupt mode says whether CPU interrupt occurs at start or end of transfer

MODE Register – cont.

15	14	13	12	11	10	9	8
CHINTE	DATASIZE	SYNCSEL	SYNCE	CONTINUOUS	ONESHOT	CHINTMODE	PERINTE
R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0

7	6	5	4	0
OVRINTE	Reserved		PERINTSEL	
R/W - 0	R/W - 0		R/W - 0	

PERINTE	Peripheral interrupt enable allows peripheral interrupt event to start transfer
OVRINTE	Overrun interrupt enable allows the OVRFLG bit to generate a CPU interrupt
PERINTSEL	Peripheral interrupt select field determines which peripheral can start a transfer

CONTROL Register

15	14	13	12	11	10	9	8
Reserved	OVRFLG	RUNSTS	BURSTSTS	TRANSFERSTS	SYNCERR	SYNCFLG	PERINTFLG
R - 0	R - 0	R - 0	R - 0	R - 0	R - 0	R - 0	R - 0

7	6	5	4				0
ERRCLR	SYNCCLR	SYNCFRC	PERINTCLR	PERINTFRC	SOFTRESET	HALT	RUN
R=0/W=1	R=0/W=1	R=0/W=1	R=0/W=1	R=0/W=1	R=0/W=1	R=0/W=1	R=0/W=1

OVRFLG	Indicates an overrun condition
RUNSTS	Indicates the channel is enabled
BURSTSTS	Indicates the channel is in the middle of a burst
TRANSFERSTS	Indicates the channel is in the middle of a transfer
SYNCERR	Indicates a sync error has occurred
SYNCFLG	Indicates a sync event has occurred
PERINTFLG	Indicates a peripheral event has occurred. This is cleared as soon as the burst starts.

CONTROL Register – cont.

15	14	13	12	11	10	9	8
Reserved	OVRFLG	RUNSTS	BURSTSTS	TRANSFER STS	SYNCERR	SYNCFLG	PERINTFLG
R - 0	R - 0	R - 0	R - 0	R - 0	R - 0	R - 0	R - 0
7	6	5	4				0
ERRCLR	SYNCCLR	SYNCFRC	PERINTCLR	PERINTFRC	SOFTRESET	HALT	RUN
R=0/W=1	R=0/W=1	R=0/W=1	R=0/W=1	R=0/W=1	R=0/W=1	R=0/W=1	R=0/W=1

ERRCLR	Clears both the SYNCERR and the OVRFLG bits
SYNCCLR	Clears the SYNCFLG bit
SYNCFRC	Forces a sync event to occur
PERINTCLR	Clears the PERINTFLG bit
PERINTFRC	Forces a peripheral event to occur regardless of PERINTSEL bit field status
SOFTRESET	Resets the channel's state machine
HALT	Halts the channel's state machine at one of three predefined points
RUN	Sets the RUNSTS bit and enables the channel for operation

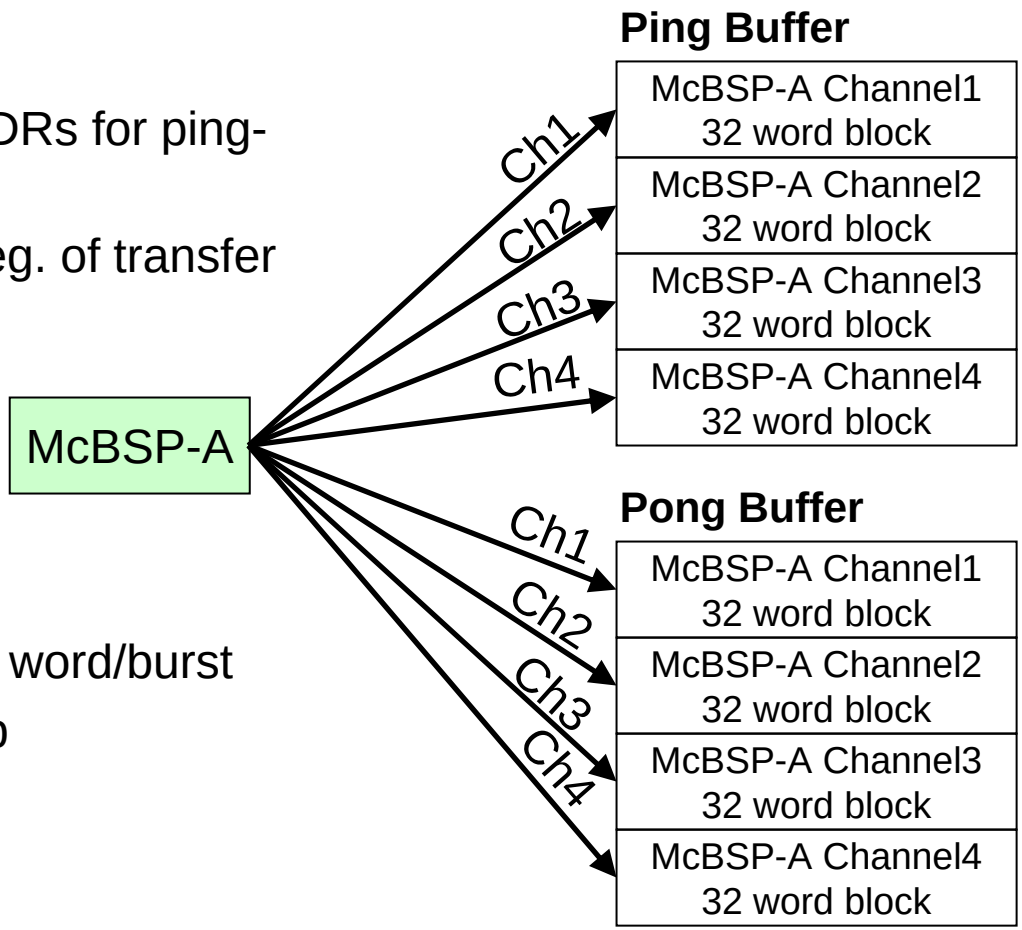
Agenda

- ✓ F2833x Block Diagram/Overview
- ✓ Channel Capabilities
- ✓ State Machine
- ✓ Address Control
- ✓ Other Features/Register Descriptions
- ❑ **Examples**

Scenario 1 – McBSP Ping-Pong

- Use shadowed ADDRs for ping-pong buffers
- CPU interrupt @ beg. of transfer

`Ch3, Ch2, Ch1,`



- McBSP is always 1 word/burst
- Bin data using wrap

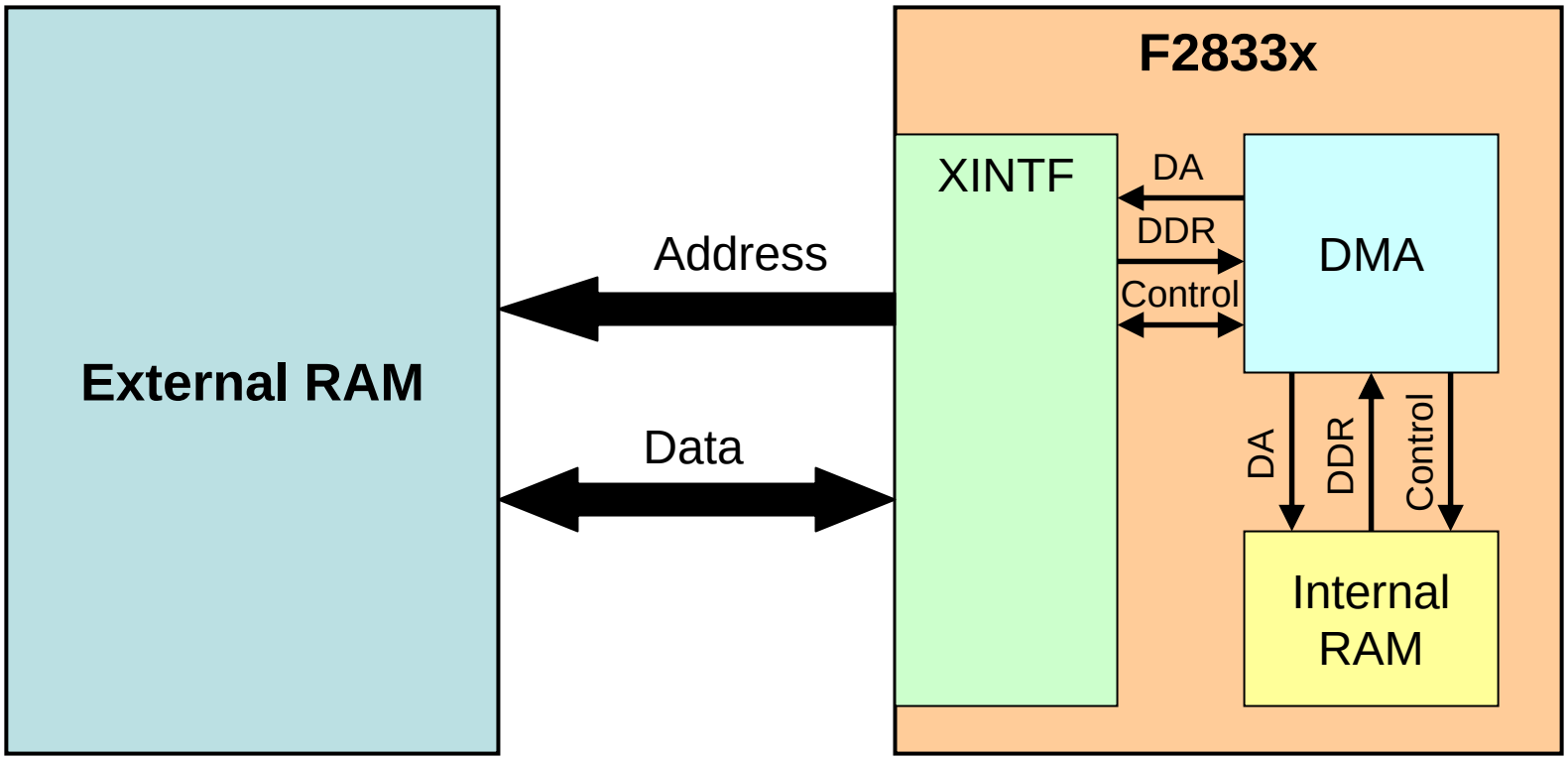
Scenario 1 – McBSP Ping-Pong

```
MODE.CH1[PERINTSEL] = MREVTA;
MODE.CH1[OVRINTE] = OVERRUN_INT_ENABLED;
MODE.CH1[PERINTE] = PER_INT_ENABLED;
MODE.CH1[CHINTMODE] = INT_AT_BEG_OF_XFER;
MODE.CH1[ONESHOT] = WAIT_FOR_EVENT;
MODE.CH1[CONTINUOUS] = KEEP_ENABLED_AT_END;
MODE.CH1[SYNCE] = SYNC_DISBALED;
MODE.CH1[DATASIZE] = 16BIT_XFERS;
MODE.CH1[CHINTE] = CPU_INT_ENABLED;
BURST_SIZE = 0;
TRANSFER_SIZE = 127;
SRC_ADDR.shadow = 0x00005001;
SRC_WRAP_SIZE = 0xFFFF;
SRC_BURST_STEP = 0;
SRC_TRANSFER_STEP = 0;
DST_ADDR.shadow = 0x0000E000;
DST_BEG_ADDR.shadow = 0x0000E000;
DST_WRAP_SIZE = 3;
DST_TRANSFER_STEP = 32;
DST_WRAP_STEP = 1;
CONTROL.CH1[RUN] = CHANNEL_ENABLED;
```

```
interrupt void DMA_CH1_ISR(void)
{
  if (CONTROL.CH1[OVRFLG] == 1)
  {
    HandleError();
  }
  EALLOW;
  if (Buffer == PING)
  {
    DST_ADDR.shadow = 0x00F000;
    DST_BEG_ADDR.shadow = 0xF000;
    Buffer = PONG;
  } else
  {
    DST_ADDR.shadow = 0x00E000;
    DST_BEG_ADDR.shadow = 0xE000;
    Buffer = PING;
  }
  EDIS;
  if (!FirstInterrupt)
  {
    Process_Data();
  }
  EnableDMACH7();
}
```

Value	Interrupt	Peripheral
0	none	No Peripheral Connection
1	SEQ1INT	ADC
2	SEQ2INT	
3	XINT1	External Interrupts
4	XINT2	
5	XINT3	
6	XINT4	
7	XINT5	
8	XINT6	
9	XINT7	
10	TINT0	CPU Timers
11	TINT1	
12	TINT2	
13	MXEVTA	McBSP-A
14	MREVTA	
15	MXEVTB	McBSP-B
16	MREVTB	
17	ADCEVT	ADC
31:18	spare	No peripheral connection.

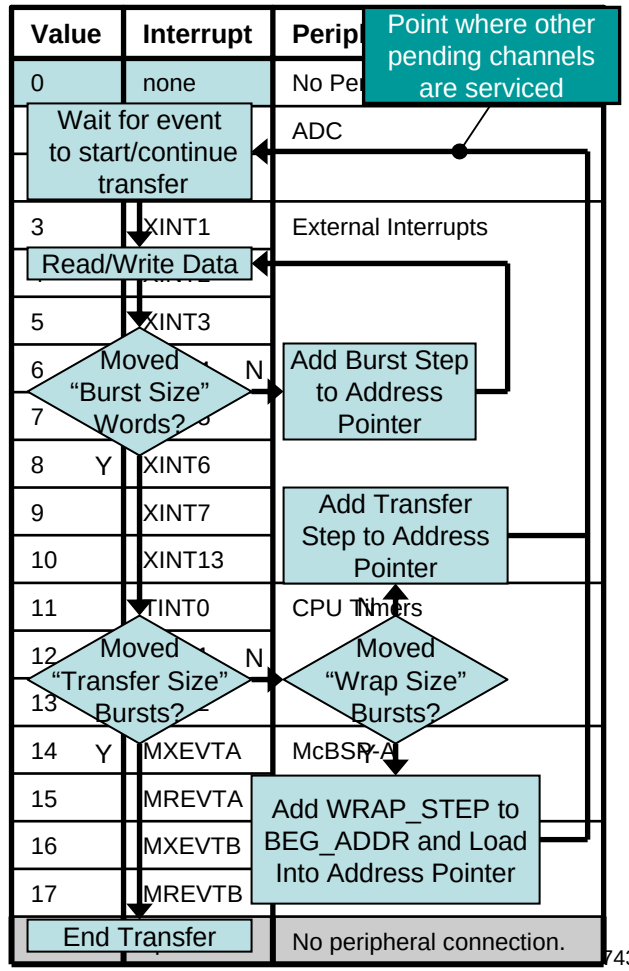
Scenario 2 – 32 Bit External Memory



Move 1k x 32 block from external memory.

Scenario 2 – 32 Bit External Memory

```
MODE.CH1[PERINTSEL] = 0;
MODE.CH1[OVRINTE] = OVERRUN_INT_DISABLED;
MODE.CH1[PERINTE] = PER_INT_ENABLED;
MODE.CH1[CHINTMODE] = INT_AT_END_OF_XFER;
MODE.CH1[ONESHOT] = COMPLETE_ENTIRE_XFER;
MODE.CH1[CONTINUOUS] = DISABLE_AT_END;
MODE.CH1[SYNCE] = SYNC_DISBALED;
MODE.CH1[DATASIZE] = 32BIT_XFERS;
MODE.CH1[CHINTE] = CPU_INT_ENABLED;
BURST_SIZE = 31;
TRANSFER_SIZE = 63;
SRC_ADDR.shadow = 0x00004000;
SRC_WRAP_SIZE = 0xFFFF;
SRC_BURST_STEP = 2;
SRC_TRANSFER_STEP = 2;
DST_ADDR.shadow = 0x0000C000;
DST_WRAP_SIZE = 0xFFFF;
DST_BURST_STEP = 2;
DST_TRANSFER_STEP = 2;
CONTROL.CH1[RUN] = CHANNEL_ENABLED;
CONTROL.CH1[PERINTFRC] = 1;
```



Agenda

- ✓ F2833x Block Diagram/Overview
- ✓ Channel Capabilities
- ✓ State Machine
- ✓ Address Control
- ✓ Other Features/Register Descriptions
- ✓ Examples

Highlights Summary

- DMA is event driven and contains 6 channels w/ 74 possible input sources (software, timers, McBSP's, ADC, external pins)
- Transfers are made up of multiple bursts, switching between channels is done between these bursts
- Address manipulation is flexible and can be used to rearrange data as it is transferred – useful in binning, transposing matrices, reversing order of array data, etc.
- An overrun flag/interrupt is provided to catch conditions where the system is overloading the DMA
- Throughput is ~4 or 5 cycles/word (16-bit or 32-bit word)
- Channel 1 can be made a higher priority than the others
- Ping pong buffers are easy w/ shadowed address pointers
- Frees up CPU resources!

Thank You

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Clocks and Timers	www.ti.com/clocks	Digital Control	www.ti.com/digitalcontrol
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
RFID	www.ti-rfid.com	Telephony	www.ti.com/telephony
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265