Lecture 4
- What are interrupts?
- What is CSL and BSL?
- Using AD535 codec
  - Output sine to DSK codec via serial port
- Using Interrupts
  - Use serial port interrupt to synchronize sinewave sample output

Interruptions
- An event that causes processor to halt what it is doing, and execute an interrupt service routine (ISR)
- Sources of interrupts include:
  - Timers
  - External interrupts
  - DMA (direct memory access)
  - McBSP transmit or receive

Interruptions
- Interrupt event to CPU interrupts
- Create the interrupt vector table

Interrupt service routines
1. Save registers (context save)
2. Actual function to perform
3. Restore registers (context restore)

Enabling interrupts
- Individual interrupts can be turned on or off (set appropriate bits in Interrupt Enable Register)
- A master switch can be used to turn all interrupts on or off (Global Enable Interrupt bit in the Control Status Register)

TI Software Foundation Libraries
Board Support Library (BSL)
Higher-level routines supporting DSK-specific functionality
BSL routines make use of CSL routines
- bsl_ad535.h
- bsl_led.h
- bsl_dip.h
- ...

Chip Support Library (CSL)
Low-level routines supporting on-chip peripherals
- mcbsp.h
- edm.h
- irq.h
- ...

C Compiler
Runtime Support Library
Standard ANSI C libraries
- stdio.h
- math.h
- ...

TI DSP
Peripheral Support Libraries

<table>
<thead>
<tr>
<th>'C6000 CSL Modules</th>
<th>'C6711 DSK CSL</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSL</td>
<td>CSL initialization</td>
</tr>
<tr>
<td>CHIP</td>
<td>BOARD specify board</td>
</tr>
<tr>
<td>CACHE</td>
<td>Cache</td>
</tr>
<tr>
<td>DAT</td>
<td>DIP Access audio codec</td>
</tr>
<tr>
<td>DMA</td>
<td>DIP Readboard DIP switches</td>
</tr>
<tr>
<td>EDMA</td>
<td>FLASH Program FLASH ROM</td>
</tr>
<tr>
<td>EMIF</td>
<td>LED Write LED's</td>
</tr>
<tr>
<td>HIPI</td>
<td>DSP/BIOS</td>
</tr>
<tr>
<td>IRQ</td>
<td>HWI Hardware interrupts</td>
</tr>
<tr>
<td>MCBP</td>
<td>Multichannel buffered serial port</td>
</tr>
<tr>
<td>PWR</td>
<td>Power down</td>
</tr>
<tr>
<td>STD/C</td>
<td>Standard include</td>
</tr>
<tr>
<td>Timer</td>
<td></td>
</tr>
</tbody>
</table>

General Procedure to use BSL/CSL

1. Declare variables
   - Usually handle & configure

   Example: AD535_Handle myHandle;
   AD535_Config myConfig = { AD535_LOOPBACK_DISABLE, ...
   main() { myHandle = AD535_open(AD535_localId); }

2. Open peripheral
   - Reserved resource
   - Provides 'handle' to reference resource
   (AD535_localId is a pre-defined pointer, special for the codec on the DSK)

   Example: AD535_Handle myHandle;
   AD535_Config myConfig = { AD535_LOOPBACK_DISABLE, ...
   main() { myHandle = AD535_open(AD535_localId); }

3. Configure peripheral
   - Apply setup configuration to peripheral

   Example: AD535_Handle myHandle;
   AD535_Config myConfig = { AD535_LOOPBACK_DISABLE, ...
   main() { myHandle = AD535_open(AD535_localId); }
   AD535_config(myHandle, &myConfig); }

4. Use peripheral
   - Some patch's use functions to read/write items

   Example: AD535_Handle myHandle;
   AD535_Config myConfig = { AD535_LOOPBACK_DISABLE, ...
   main() { myHandle = AD535_open(AD535_localId); }
   AD535_config(myHandle, &myConfig); }
   AD535_write(myHandle, value); }

General Procedure to use BSL/CSL

1. Declare variables
   - Usually handle & configure

   Example: AD535_Handle myHandle;
   AD535_Config myConfig = { AD535_LOOPBACK_DISABLE, ...
   main() { myHandle = AD535_open(AD535_localId); }

2. Open peripheral
   - Reserved resource
   - Provides 'handle' to reference resource

3. Configure peripheral
   - Apply setup configuration to peripheral

4. Use peripheral
   - Some patch's use functions to read/write items

Creating a Sine Wave

Sine_gen.c
Generates a value for each output sample
Output Sine via AD535 Codec

Send to Codec via McBSP

Codec

McBSP Block Diagram

AD535_write

Synchronizing the Output Samples

Algorithm

Generates a single sine sample

Timing

How often should we output a sample?

Since sineGen is faster than 1 ms, how do we wait another 124+ us?

McBSP transmit interrupt

AD535 Config
- structure used to configure AD535 parameters such as input/output gains, etc.
- AD535_open() reserves resource and returns handle
- AD535_config() configures AD535 and McBSP0
- AD535_read() reads from AD535 ADC thru McBSP0 receive
- AD535_write() writes to AD535 DAC thru McBSP0 transmit

McBSP Control Registers

CPU

Sine Gen

° ± ²

Output Sine via AD535 Codec

Send to Codec via McBSP

Codec

McBSP Block Diagram

AD535_write

Synchronizing the Output Samples

Algorithm

Generates a single sine sample

Timing

How often should we output a sample?

Since sineGen is faster than 1 ms, how do we wait another 124+ us?

McBSP transmit interrupt

AD535 Config
- structure used to configure AD535 parameters such as input/output gains, etc.
- AD535_open() reserves resource and returns handle
- AD535_config() configures AD535 and McBSP0
- AD535_read() reads from AD535 ADC thru McBSP0 receive
- AD535_write() writes to AD535 DAC thru McBSP0 transmit

McBSP Control Registers

CPU

Sine Gen

° ± ²
Real-time Sync to XINT0

Codec Initialization

Enabling Interrupts

Turning Function into an ISR

Codec Initialization

Enabling Interrupts
Enabling Interrupts

'C6711

- Algorithm
- Real-time events
- Initialization

```
DHW
// Remember, we used HWI for McBSP transmit int (X INTO)
IRQ_MAILUROEVT_XINTO, 0; /* CSL irq, h */
IRQ_EnableIRQ_EVT_XINTO();

// Enable interrupts globally (this sets the GIE bit in CSR if HWI enabled); */ API Function, Config Tool*/
main()
(CSL_Init();
BL_Init();
codec_Init();
init_HW();
ADSS_write(ADDS5, 0);
while(1);
codec_out(); /* XINTO_HW */
{ ADSS_write(ADDS5, sineGen());
}

```

Enabling Interrupts

'C6711

- Algorithm
- Real-time events
- Initialization

```
DHW
// Output a sample to start generating interrupts
main()
(CSL_Init();
BL_Init();
codec_Init();
init_HW();
ADSS_write(ADDS5, 0);
while(1);
codec_out(); /* XINTO_HW */
{ ADSS_write(ADDS5, sineGen());
}
```