TI DSP C2000 Serial

TI DSP C2000
Content

- 1 DSP introduction
- 2 TI TMS320FL2407 (as 2407)
- 3 2812 (opt.)
DSP introduction

DSP Market Share 2003

source: Forward Concepts, 2004

- Texas Instruments: 48.00%
- Agere Systems: 19%
- Motorola: 13%
- Analog Devices: 10%
- Other: 10%

Total Revenue: 6,130 Million US-$

www.fwdconcepts.com
TI DSP introduction

TMS320 – Family Branches

C2000
Efficient Integration for Control
DSC

C5000
Power Efficient Performance
DSP

C6000
High Performance ‘C’ Efficiency
DSP
TI TMS320 family

- Different families and sub-families exist to support different markets.

- **Lowest Cost**
  - Control Systems
    - Motor Control
    - Storage
    - Digital Ctrl Systems

- **Efficiency**
  - Best MIPS per Watt / Dollar / Size
    - Wireless phones
    - Internet audio players
    - Digital still cameras
    - Modems
    - Telephony

- **Performance & Best Ease-of-Use**
  - Multi Channel and Multi Function App’s
  - Comm Infrastructure
  - Wireless Base-stations
  - Imaging
  - Multi-media Servers
  - Video
Roadmap of TMS320C2000

Future of Control: Improved Industrial Drive, Improved System Density

High-Precision Uni-processor Control for Applications from Industrial Drives to Automotive

Multi-Function, Appliance & Consumer Control

F24x
LF240xA
C24x
LC240xA

F2810 150 MIPS
F2810 150 MIPS
F2811 150 MIPS
F2811 150 MIPS
C2810 150 MIPS
C2810 150 MIPS
C2811 150 MIPS
C2811 150 MIPS
C2812 150 MIPS
C2812 150 MIPS
R2812 150 MIPS
R2812 150 MIPS
F2801 100 MIPS
F2801 100 MIPS
F2806 100 MIPS
F2806 100 MIPS
F2808 100 MIPS
F2808 100 MIPS

Samples December 04

Higher performance
Greater integration

Software Compatible

In Silicon
Announced
Optical Networking
Control of laser diode

Digital Power Supply
Provides control, sensing, PFC, and other functions

Printer
Print head control
Paper path motor control

Non-traditional Motor Control
Many new cool applications to come

Evaluating Other Segments
e.g., Musical Instruments

Broad C28x™ Application Base
## TI C2000: Portfolio for Embedded Applications

<table>
<thead>
<tr>
<th></th>
<th>F2812</th>
<th>F2810</th>
<th>LF2407A</th>
<th>LF2406A</th>
<th>LF2403A</th>
<th>LF2402A</th>
<th>LF2401A</th>
<th>LC2406A</th>
<th>LC2404A</th>
<th>LC2402A</th>
<th>LC2401A</th>
<th>F243</th>
<th>F241</th>
<th>F240</th>
<th>C242</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CPU</strong></td>
<td>32bit</td>
<td>32bit</td>
<td>16 bit</td>
<td>16 bit</td>
<td>16 bit</td>
<td>16 bit</td>
<td>16 bit</td>
<td>16 bit</td>
<td>16 bit</td>
<td>16 bit</td>
<td>16 bit</td>
<td>16 bit</td>
<td>16 bit</td>
<td>16 bit</td>
<td>16 bit</td>
</tr>
<tr>
<td><strong>MIPS</strong></td>
<td>150</td>
<td>150</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td><strong>RAM (words)</strong></td>
<td>18K</td>
<td>18K</td>
<td>2.5K</td>
<td>2.5K</td>
<td>1.0K</td>
<td>1.0K</td>
<td>1.0K</td>
<td>1.5K</td>
<td>544</td>
<td>1.0K</td>
<td>544</td>
<td>544</td>
<td>544</td>
<td>544</td>
<td>544</td>
</tr>
<tr>
<td><strong>ROM (words)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>32K</td>
<td>16K</td>
<td>6K</td>
<td>8K</td>
</tr>
<tr>
<td><strong>Flash (words)</strong></td>
<td>128K</td>
<td>64K</td>
<td>32K</td>
<td>32K</td>
<td>16K</td>
<td>8K</td>
<td>8K</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>8K</td>
<td>8K</td>
<td>16K</td>
<td></td>
</tr>
<tr>
<td><strong>BootROM (words)</strong></td>
<td>4K</td>
<td>4K</td>
<td>256</td>
<td>256</td>
<td>256</td>
<td>256</td>
<td>256</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Event Manager</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>PWM(CMP)</strong></td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>8</td>
<td>8</td>
<td>7</td>
<td>16</td>
<td>16</td>
<td>8</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>12</td>
<td>8</td>
</tr>
<tr>
<td><strong>TIMER</strong></td>
<td>7</td>
<td>7</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td><strong># ofChan</strong></td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>8</td>
<td>8</td>
<td>5</td>
<td>16</td>
<td>16</td>
<td>8</td>
<td>5</td>
<td>8</td>
<td>8</td>
<td>16</td>
<td>8</td>
</tr>
<tr>
<td><strong>Conv time</strong></td>
<td>200ns</td>
<td>200ns</td>
<td>500ns</td>
<td>500ns</td>
<td>500ns</td>
<td>500ns</td>
<td>500ns</td>
<td>375ns</td>
<td>375ns</td>
<td>425ns</td>
<td>500ns</td>
<td>900ns</td>
<td>900ns</td>
<td>0.1us</td>
<td>0.1us</td>
</tr>
<tr>
<td><strong>McBSP</strong></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>EXMIF</strong></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Watch Dog</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>SPI</strong></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>SCI (UART)</strong></td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><strong>CAN</strong></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Volts (V)</strong></td>
<td>1.8 core 3.3 I/O</td>
<td>1.8 core 3.3 I/O</td>
<td>3.3</td>
<td>3.3</td>
<td>3.3</td>
<td>3.3</td>
<td>3.3</td>
<td>3.3</td>
<td>3.3</td>
<td>3.3</td>
<td>5.0</td>
<td>5.0</td>
<td>5.0</td>
<td>5.0</td>
<td></td>
</tr>
<tr>
<td><strong># I/O</strong></td>
<td>56</td>
<td>41</td>
<td>41</td>
<td>21</td>
<td>21</td>
<td>13</td>
<td>41</td>
<td>41</td>
<td>21</td>
<td>13</td>
<td>32</td>
<td>26</td>
<td>28</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td><strong>Package</strong></td>
<td>176LQFP</td>
<td>128LQFP</td>
<td>144LQFP</td>
<td>100LQFP</td>
<td>64LQFP</td>
<td>64PQFP</td>
<td>32LQFP</td>
<td>100LQFP</td>
<td>100LQFP</td>
<td>64PQFP</td>
<td>32LQFP</td>
<td>144LQFP</td>
<td>64PQFP</td>
<td>68PLCC</td>
<td>132PQFP</td>
</tr>
</tbody>
</table>

2008-9-18
College of Electrical Engineering, Zhejiang University
INTRODUCTION TO THE
TMSLF2407 DSP CONTROLLER

TI DSP C2000
Introduction

- The Texas Instruments TMS320LF2407 DSP Controller (LF2407) is a programmable digital controller with a C2xx DSP central processing unit (CPU) as the core processor.

- The LF2407 contains the DSP core processor and useful peripherals integrated onto a single piece of silicon.

- The LF2407 combines the powerful CPU with on-chip memory and peripherals.

- With the DSP core and control-oriented peripherals integrated into a single chip, users can design very compact and cost-effective digital control systems.
Introduction

- The LF2407 DSP controller offers 40 million instructions per second (MIPS) performance.
- This high processing speed of the C2xx CPU allows users to compute parameters in real time rather than look up approximations from tables stored in memory.
- This fast performance is well suited for processing control parameters in applications such as notch filters or sensorless motor control algorithms where a large amount of calculations must be computed quickly.
Introduction

- The LF2407 peripheral set includes:
  - Two Event Managers (A and B)
  - General Purpose (GP) timers
  - PWM generators for digital motor control
  - Analog-to-digital converter
  - Controller Area Network (CAN) interface
  - Serial Peripheral Interface (SPI) ?synchronous serial port
  - Serial Communications Interface (SCI) ?asynchronous serial port
  - General-Purpose bi-directional digital I/O (GPIO) pins
  - Watchdog Timer (time-out? DSP reset device for system integrity)
peripheral
Indicates optional modules in the 240x family. The memory size and peripheral selection of these modules change for different 240xA devices.
Event Managers (EVA, EVB)

- There are two Event Managers on the LF2407, the EVA and EVB.
- The Event Manager is the **most important peripheral** in digital motor control. It contains the necessary functions needed to control electromechanical devices.
- Each EV is composed of functional ‘blocks’ including:
  - timers,
  - comparators,
  - capture units for triggering on an event,
  - PWM logic circuits,
  - quadrature-encoder-pulse (QEP) circuits,
  - interrupt logic.
The Analog-to-Digital Converter (ADC)

- The ADC on the LF2407 is used whenever an external analog signal needs to be sampled and converted to a digital number.

- Examples of ADC applications range from sampling a control signal for use in a digital notch filtering algorithm or using the ADC in a control feedback loop to monitor motor performance.

- Additionally, the ADC is useful in motor control applications because it allows for current sensing using a shunt resistor instead of an expensive current sensor.
The Control Area Network (CAN) Module

- The CAN module is used for multi-master serial communication between external hardware.

- The CAN bus has a high level of data integrity and is ideal for operation in noisy environments such as in an automobile, or industrial environments that require reliable communication and data integrity.
Serial Peripheral Interface (SPI)

- The SPI is a high-speed synchronous communication port that is mainly used for communicating between the DSP and external peripherals or another DSP device.

- Typical uses of the SPI include communication with external shift registers, display drivers, or ADCs.
Serial Communications Interface (SCI)

- The SCI is an asynchronous communication port that supports asynchronous serial (UART) digital communication between the CPU and other asynchronous peripherals that use the standard NRZ (non-return-to-zero) format.

- It is useful in communication between external devices and the DSP.

- Since these communication peripherals are not directly related to motion control applications, they will not be discussed further in this text.
The Watchdog timer (WD) peripheral monitors software and hardware operations and asserts a system reset when its internal counter overflows.

The WD timer (when enabled) will count for a specific amount of time.

It is necessary for the user’s software to reset the WD timer periodically so that an unwanted reset does not occur.

If for some reason there is a CPU disruption, the watchdog will generate a system reset.
Watchdog Timer (WD)

- For example, if the software enters an endless loop or if the CPU becomes temporarily disrupted, the WD timer will overflow and a DSP reset will occur, which will cause the DSP program to branch to its initial starting point.

- Most error conditions that temporarily disrupt chip operation and inhibit proper CPU function can be cleared by the WD function.

- In this way, the WD increases the reliability of the CPU, thus ensuring system integrity.
General Purpose Bi-Directional Digital I/O (GPIO) Pins

- Since there are only a finite number of pins available on the LF2407 device, many of the pins are multiplexed to either their primary function or the secondary GPIO function.

- In most cases, a pin’s second function will be as a general-purpose input/output pin.

- The GPIO capability of the LF2407 is very useful as a means of controlling the functionality of pins and also provides another method to input or output data to and from the device.
The JTAG port provides a standard method of interfacing a personal computer with the DSP controller for emulation and development.

The XDS510PP or equivalent emulator pod provides the connection between the JTAG module on the LF2407 and the personal computer.

The JTAG module allows the PC to take full control over the DSP processor while Code Composer Studio™ is running.
Joint Test Action Group (JTAG) Port

- Figure shows the connection scheme from computer to the DSP board.
Phase Locked Loop (PLL) Clock Module

- The phase locked loop (PLL) module is basically an input clock multiplier that allows the user to control the input clocking frequency to the DSP core.

- External to the LF2407, a clock reference (can oscillator/crystal) is generated.

- This signal is fed into the LF2407 and is multiplied or divided by the PLL. This new (higher or lower frequency) clock signal is then used to clock the DSP core.
Software Tools

- Texas Instrument’s Code Composer Studio™ (CCS) is a user-friendly Windows-based debugger for developing and debugging software for the LF2407.

- CCS allows users to write and debug code in C or in TI assembly language.

- CCS has many features that can aid in developing code.
Software Tools

- CCS features include:
  - User-friendly Windows environment
  - Ability to use code written in C and assembly
  - Memory displays and on-the-fly editing capability
  - Disassembly window for debugging
  - Source level debugging, which allows stepping through and setting breakpoints in original source code
  - CPU register visibility and modification
  - Real-time debugging with watch windows and continuous refresh
  - Various single step/step over/step-into command icons
  - Ability to display data in graph formats
  - General Extension Language (GEL) capability, allows the user to create functions that extend the usefulness of CCSTM
TMSLF2407 DSP CONTROLLER

A first glance
TMS320C240x Block Diagram
Multiplier and ALU / Shifters
Program Memory
Data Memory
TMS320C240x Memory Map

<table>
<thead>
<tr>
<th>Hex</th>
<th>Program</th>
<th>Hex</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>Interrupts</td>
<td>0000</td>
<td>Memory-Mapped</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Registers</td>
</tr>
<tr>
<td>0040</td>
<td>On-chip ROM / Flash (External if MP/MC = 1)</td>
<td>0060</td>
<td>On-chip DARAM B2</td>
</tr>
<tr>
<td>X</td>
<td>External</td>
<td>0080</td>
<td>Reserved</td>
</tr>
<tr>
<td>8000</td>
<td>SARAM (2K) (PON = 1) or External (PON = 0)</td>
<td>0200</td>
<td>On-chip DARAM B0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(CNF = 0) or</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Reserved (CNF = 1)</td>
</tr>
<tr>
<td>8800</td>
<td>External</td>
<td>0300</td>
<td>On-chip DARAM B1</td>
</tr>
<tr>
<td>FE00</td>
<td>Reserved (CNF = 1)</td>
<td>0400</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>External (CNF = 0)</td>
<td>0800</td>
<td>SARAM (2K) (DON = 1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>or Reserved (DON = 0)</td>
</tr>
<tr>
<td>FF00</td>
<td>On-chip DARAM B0 (CNF = 1) or External (CNF = 0)</td>
<td>7000</td>
<td>Non-EV Peripherals</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>EV Peripherals</td>
</tr>
<tr>
<td>FFFF</td>
<td></td>
<td>7400</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>External</td>
</tr>
</tbody>
</table>
## Pipeline Process

<table>
<thead>
<tr>
<th>Cycle</th>
<th>100</th>
<th>101</th>
<th>102</th>
<th>103</th>
<th>104</th>
<th>105</th>
<th>106</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>F₁</td>
<td>D₁</td>
<td>R₁</td>
<td>E₁</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sub</td>
<td>F₂</td>
<td>D₂</td>
<td>R₂</td>
<td>E₂</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mpy</td>
<td>F₃</td>
<td>D₃</td>
<td>R₃</td>
<td>E₃</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Store</td>
<td>F₄</td>
<td>D₄</td>
<td>R₄</td>
<td>E₄</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Fully loaded pipeline (normal operation)*
Peripherals

Event Manager
- GP Timers
- Compare Unit
- PWM Outputs
- Dead-Band Logic
- Capture Unit
- Quadrature Encoder Pulse (QEP)

Non-EV Manager
- Watchdog Timer
- SPI
- SCI
- A/D Converter
- I/O Pins
- CAN

Data Bus
TMS320C240x Instruction Set

Accumulator Memory Reference

| ABS | NEG | SUB |
| ADD | NORM | SUBB |
| ADDC | OR | SUBC |
| ADDS | ROL | SUBS |
| ADDT | ROR | SUBT |
| AND | SACH | XOR |
| CMPL | SACL | ZALR |
| LACC | SFL | |
| LACL | SFR | |
| LACT | |

Auxiliary Register and Data Page Pointer

| ADRK | MAR |
| CMPR | SAR |
| LAR | SBRK |
| LDP | |

Multiply, T, P

| APAC | MPYA |
| LPH | MPYS |
| LT | MPYU |
| LTA | PAC |
| LTD | SPAC |
| LTP | SPH |
| LTS | SPL |
| MAC | SPM |
| MACD | SQRA |
| MPY | SQRS |

Control Instructions

| BIT | POP |
| BITT | POPD |
| CLRC | PSHD |
| IDLE | PUSH |
| RPT | SETC |
| LST | SST |
| NOP | |

I/O and Data Memory Operations

| BLDD | OUT |
| BLPD | SPLK |
| DMOV | TBLR |
| IN | TBLW |

Branch

| B | CC |
| BACC | INTR |
| BANZ | NMI |
| BCND | RET |
| CALA | RETC |
| CALL | TRAP |

SUN Dan
College of Electrical Engineering, Zhejiang University
TMS320LF2407 Review
Architecture

- 3.3V
- 33ns instruction cycle time 30 MIPS (40 MIPS option)
- 32K words of on-chip FLASH memory (30 MIPS)
- Flash sectoring and security
- Boot ROM 256 words
- Motor control optimized Event Manager (2x C242 like)
- A/D converter (16 inputs) with 500ns conversion time
- UART (SCI)
- SPI
- On-chip CAN module 2.0B
- 16-bit external memory interface
- up to 37 GPIO Pins
### Memory Map

#### Interrupts
- On-chip ROM / Flash
  - External if MP/MP = 1
- External
  - X: 8000 - LC/F2406, LF2407
  - X: 4000 - F/C210, LC2404
  - X: 2000 - F241, F243, LF2402
  - X: 1000 - C242, LC2402

#### Data
- Memory-Mapped Registers
  - On-chip DARAM B2
  - Reserved
  - On-chip DARAM B0
    - (CNF = 0) or
    - Reserved (CNF = 1)
  - On-chip DARAM B1
    - Reserved
  - SARAM (2K)
    - (DON = 1) or
    - Reserved (DON = 0)
- Non-EV Peripherals
  - EV Peripherals
  - Reserved
  - External
Memory Map

- The 'LF2407 memory space is divided into three regions:
  - 64K program
  - 64K data
  - 64K I/O

- The 'LF2407 has a minimum of 544 words of on-chip RAM, which is sometimes referred to as dual-access RAM. It can implement the action of a delay line without the need for “circular buffering” as any other memory would.

- On reset, this memory is found in data space, but a portion of it may be relocated under software control to program space.

- The second on-chip memory is ROM, which is located at the beginning of program space. It may be used or bypassed under control of the MP/ MC signal line. The amount of on-chip non-volatile memory (ROM or flash) varies based on the ‘LF2407 device.
Peripherals

- The ‘LF2407 devices contain peripherals optimized for motor/motion control applications.

- For many systems, this devices provide a low-cost, high performance solution.
Digital I/O Pin & Register Structure

- PORT A:
  - IOPA0 / SCITXD
  - IOPA1 / SCIRXD
  - IOPA2 / XINT1
  - IOPA3 / CAP1 / QEP1
  - IOPA4 / CAP2 / QEP2
  - IOPA5 / CAP3
  - IOPA6 / PWM1
  - IOPA7 / PWM2

- PORT B:
  - IOPB0 / PWM3
  - IOPB1 / PWM4
  - IOPB2 / PWM5
  - IOPB3 / PWM6
  - IOPB4 / T1PWM / T1CMP
  - IOPB5 / T2PWM / T2CMP
  - IOPB6 / TDIRA
  - IOPB7 / TCLKINA

- PORT C:
  - WR / IOPC0
  - BIO / IOPC1
  - IOPC2 / SPI/SIMO
  - IOPC3 / SPI/SO
  - IOPC4 / SPI/CLK
  - IOPC5 / SPI/ST
  - IOPC6 / CANTX
  - IOPC7 / CANRX

- PORT D:
  - IOPD0 / XINT2 / ADCSOC
  - EMU0 / --
  - EMU1 / --
  - TCK / --
  - TDI / --
  - TDO / --
  - TMS / --
  - TMS2 / --

- PORT E:
  - CLKOUT / IOPE0
  - IOPE1 / PWM7
  - IOPE2 / PWM8
  - IOPE3 / PWM9
  - IOPE4 / PWM10
  - IOPE5 / PWM11
  - IOPE6 / PWM12
  - IOPE7 / CAP4 / QEP3

- PORT F:
  - IOPF0 / CAP5 / QEP4
  - IOPF1 / CAP6
  - IOPF2 / T3PWM / T3CMP
  - IOPF3 / T4PWM / T4CMP
  - IOPF4 / TDIRB
  - IOPF5 / TCLKINB
  - IOPF6 / IOPF6

Note: label names on left are pin functions at reset.
Digital I/O Port Registers

<table>
<thead>
<tr>
<th>Address</th>
<th>Register</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>7090h</td>
<td>MCRA</td>
<td>I/O Mux Control Register A</td>
</tr>
<tr>
<td>7092h</td>
<td>MCRB</td>
<td>I/O Mux Control Register B</td>
</tr>
<tr>
<td>7094</td>
<td>MCRC</td>
<td>I/O Mux Control Register C</td>
</tr>
<tr>
<td>7098h</td>
<td>PADATDIR</td>
<td>I/O Port A Data and Direction Register</td>
</tr>
<tr>
<td>709Ah</td>
<td>PBDAFDIR</td>
<td>I/O Port B Data and Direction Register</td>
</tr>
<tr>
<td>709Ch</td>
<td>PCDATDIR</td>
<td>I/O Port C Data and Direction Register</td>
</tr>
<tr>
<td>709Eh</td>
<td>PDDATDIR</td>
<td>I/O Port D Data and Direction Register</td>
</tr>
<tr>
<td>7095h</td>
<td>PEDATDIR</td>
<td>I/O Port E Data and Direction Register</td>
</tr>
<tr>
<td>7096h</td>
<td>PFDATDIR</td>
<td>I/O Port F Data and Direction Register</td>
</tr>
</tbody>
</table>
Watchdog Timer

- The watchdog timer provides a safeguard against CPU crashes by automatically initiating a reset if it is not serviced by the CPU at regular intervals.

- In motor control applications, this helps protect the motor and drive electronics when control is lost due to a CPU lockup.

- Any CPU reset will revert the PWM outputs to a high-impedance state, which should turn off the power converters in a properly designed system.
Watchdog Timer

- The watchdog timer is running immediately after system power-up/reset, and must be dealt with by software soon after.
- Specifically, you have 6.55 ms after any reset before a watchdog initiated reset will occur.
- For a 25 ns CPU clock, this translates into 262,000 instruction cycles, which is a seemingly tremendous amount!
- Indeed, this is plenty of time to get the watchdog configured as desired and serviced.
- A failure of your software to properly handle the watchdog after reset could cause an endless cycle of watchdog initiated resets to occur.
Watchdog Timer

- Resets the C240x if the CPU crashes
  - Watchdog counter runs independent of CPU
  - If counter overflows, reset is triggered
  - CPU must write correct data key sequence to reset the counter before overflow

- Watchdog must be serviced (or disabled) within ~6.55mS after reset

- With a 25 ns CPUCLK, 6.55mS translates into 262,000 instructions!
Watchdog Timer Control Register
WDCR @ 7029h

WD Flag Bit
Gets set when the WD causes a reset
• Writing a 1 clears this bit
• Writing a 0 has no effect

Watchdog Disable Bit
(Functional only if WD OVERRIDE
bit in SCSR2 is equal to 1)
System Control and Status Register 1
SCSR1 @ 7018h

- **CLKOUT Pin Source Select**
  - 0 = CPU clock
  - 1 = Watchdog Clock

- **PLL Clock Prescale**

- **Module Clock Enable Bit**
  - 0 = Disabled
  - 1 = Enabled

  - Gets set when an illegal address is accessed
  - An illegal address event also triggers an NMI
  - This bit is not cleared by reset
  - Writing a 1 clears this bit
  - Writing a 0 has no effect
System Control and Status Register 2
SCSR2 @ 7019h

WD Protect Bit
- After reset - bit gives user ability to disable WDDIS in WDCR
- 0 = protects WD from being disabled by s/w
- 1 = allows WD to be disabled using WDDIS bit in WDCR

Boot Enable
- 0 = Enable Boot ROM
- 1 = Disable Boot ROM

Microprocessor / Microcontroller Select

XMIF_Hi-Z Control
- 0 = normal
- 1 = Hi-Z state

SARAM Program / Data Space Select

Diagram showing the bit allocation and descriptions for each bit.

SUN Dan
College of Electrical Engineering, Zhejiang University
C2407- Interrupt System

- Interrupts provide a means of directing the 'C240x to suspend its main program in order to respond to a hardware-driven event.
- This eliminates the need to poll external events via software (unless desired), and improves response time and decrease processor overhead dramatically.
- Typically, interrupts are generated by devices which need to give or take data from the 'C240x.
- Examples of such devices are A/D and D/A converters and other processors.
C2407- Interrupt System

- Interrupt may also be used as a signal to inform the 'C240x when any event of interest has occurred within the system.
- When the 'C240x recognizes the interrupt signal, it suspends execution of the main program and begins execution of the code specific to the particular interrupt event.
- On the 'C240x, the programmer can dynamically select when interrupts may be taken, and which interrupts will be recognized.
C2407- Interrupt System

- The ‘C240x core of the ‘C240x processor supports six user-maskable interrupts.
- These interrupts are then fanned out and shared among numerous on-chip peripherals and external pins.
- Interrupts can be generated by internal or external sources or by software interrupt instructions.
- A reset function, a non-maskable interrupt, and a power-drive protection interrupt are also supported on all ‘C240x devices.
C240x Core Interrupt Lines

- 2 non-maskable interrupts (RS, NMI)
- 6 maskable interrupts (INT1 - INT6)
Interrupt Sources
Peripheral and externally generated interrupts are arranged in groups and associated with one of the core interrupts, INT1 - INT6.

When a valid signal is generated by a peripheral or by an external source, the corresponding individual flag bit is set to 1 in the control register associated with that peripheral or external interrupt.

If the individual enable bit is set for that interrupt, an interrupt request will be passed to the arbitration logic, which prioritizes all pending interrupts in its group and then sends the highest priority interrupt request to its associated core interrupt.

The individual flag bits are set regardless of the condition of the individual enable bits.
Interrupt Management
- Individual Flags and Masks

- The schematic below depicts the signal flow to core INT1.
A valid signal on a specific interrupt line causes the latch to display a "1" in the appropriate bit. If the individual and global switches are turned on, the interrupt reaches the core.
Interrupt Management
- Core Flags and Masks

- When an interrupt request is sent to the CPU core by the arbitration logic, the corresponding bit in the Interrupt Flag Register (IFR) is set.

- Flags are set regardless of the condition of the masks (IMR) or global interrupt switch (INTM).

- The core flag bits are automatically cleared by the processor after the interrupt is recognized by the CPU.
Interrupt Management
- Core Flags  IFR @ 0006h

- IFR indicates when a valid interrupt has occurred
- “1” is displayed in the corresponding bit
Interrupt Management
- Core Masks IMR @ 0004h

- The Interrupt Mask Register (IMR) allows the user to select which core interrupts the 'C240x should respond to at a given time.
- A logic 1 written to any mask bit enables the corresponding interrupt.
- Notice that RS and NMI cannot be masked.
Interrupt Management
- Core Masks

IMR @ 0004h

- IMR allows individual masking of each bit
  - 0 = disable (mask)
  - 1 = enable
Interrupt Management - Global Switch

- The INTM bit (bit 9 in ST0) can be used to globally enable or disable all maskable interrupts.

- When INTM = 0, all enabled interrupts (as indicated in the IMR register) are allowed.

- INTM = 1 inhibits these interrupts.

- Note that the IFR and IMR are not affected by the state of INTM.
Interrupt Management - Global Switch

- INTM bit is global enable/disable of interrupts
  - 0 = enabled
  - 1 = disabled

- If INTM = 0, all individually enabled interrupts in the IMR register are enabled
Interrupt Management - Review
Maskable Interrupt Recognition

ext. INT signal

INT high 1 CPUCLK
INT low 1 CPUCLK

Indiv. Flag Bit(s) Set

Indiv. Mask bit OK?

INTERNAL INT source

IFR Flag Bit(s) Set

IMR Bit(s) = 1?

INTM Bit = 0?

Interrupt Hardware Sequence

ISR
Maskable Interrupt Timeline - Review

1. Valid signal
2. Individual flag bit set
3. Individual interrupt enabled?
4. Core interrupt flag bit set
5. Core interrupt enabled?
6. Global interrupt switch enabled?
7. Interrupt hardware sequence
8. Branch to ISR (Interrupt Vector)
9. Interrupt Service Routine
# Interrupt Vector Locations

<table>
<thead>
<tr>
<th>Interrupt</th>
<th>Location (Hex)</th>
<th>Description</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>RS</td>
<td>0</td>
<td>Reset</td>
<td>1</td>
</tr>
<tr>
<td>INT1</td>
<td>2</td>
<td>Core Interrupt #1</td>
<td>4</td>
</tr>
<tr>
<td>INT2</td>
<td>4</td>
<td>Core Interrupt #2</td>
<td>5</td>
</tr>
<tr>
<td>INT3</td>
<td>6</td>
<td>Core Interrupt #3</td>
<td>6</td>
</tr>
<tr>
<td>INT4</td>
<td>8</td>
<td>Core Interrupt #4</td>
<td>7</td>
</tr>
<tr>
<td>INT5</td>
<td>A</td>
<td>Core Interrupt #5</td>
<td>8</td>
</tr>
<tr>
<td>INT6</td>
<td>C</td>
<td>Core Interrupt #6</td>
<td>9</td>
</tr>
<tr>
<td>TRAP</td>
<td>22</td>
<td>Trap Instruction Vector</td>
<td>-</td>
</tr>
<tr>
<td>NMI</td>
<td>24</td>
<td>Nonmaskable Interrupt</td>
<td>3</td>
</tr>
</tbody>
</table>

† Software interrupts such as TRAP do not have a hardware priority
Interrupt Service Routine

1. Context save
2. Determine the source of the core interrupt
3. Execute code specific to the interrupt source
4. Context restore
5. Re-enable Interrupts
6. Return to main program (RET)
Interrupt Structure of C240x
Determining Interrupt Source

- Each peripheral interrupt loads a unique offset value into the value into the *Peripheral Interrupt Vector Registers*
Multiple Interrupts / Priority

- Which interrupt is serviced first when multiple interrupts occur?

- Priority is only associated with multiple interrupts

- If any interrupt is being serviced, and global interrupts are turned on (INTM = 0), and any enabled interrupt occurs, it will be serviced immediately

- Each ISR is responsible for modifying the IMR in software to control interrupt nesting
Interrupt latency may not protect hardware when responding to over current through ISR software.

PDPINTx has a fast, clock independent logic path to high-impedance the PWM output pins (~45-55 ns).
LF2407 Event Manager
-Learning Objectives

- Pulse Width Modulation (PWM) Review

- Generate PWM with the Event Manager:
  - General-Purpose Timer
  - Compare Units

- Other Event Manager functions:
  - Capture Units
  - Quadrature Encoder Pulse (QEP) Circuit
Event Manager

- The Event Manager consists of the following blocks:
  - General-Purpose Timers
  - Full Compare Units
  - Capture Units
  - Quadrature Encoder Pulse (QEP) circuit
Event Manager Block Diagram (EVA)
General-Purpose Timers

- The GP Timers provide a time base for the operation of the compare units, and associated PWM circuits to generate PWM outputs.
- Additionally, they provide a time base for the operation of the quadrature encoder pulse (QEP) circuit (GP Timer 2 only) and the capture units.
- The GP Timers can also be used to generate a sampling period in a control system.
General-Purpose Timer Block Diagram (EVA)

Note: x = 1 or 2
General-Purpose Timer Block Diagram (EVA)

- The TxPR period register holds the user specified counting period.
- TxPR is automatically loaded from the period register buffer on a counter underflow, which is defined as TxCNT=0.
- This allows for on-the-fly timer period changes.
- Note that the period register buffer is static in that if no change in the current period value is desired, one is not required to write the same value to the buffer on successive timer cycles.
The procedure for GP Timer **Continuous-Up** Counting is as follows:

- User sets bit 6 of TxCON register high to initiate counting;
- Counting begins on next rising clock edge;
  - 1st count is a “Zero” (no increment);
  - Count up until match with period register.
Continuous-Up Counting Mode
(Used for Asymmetric PWM Waveforms)

This example:
TxCON.3-2 = 00 (reload TxCMP on underflow)
TxPR = 3
TxCMP = 1 (initially)
Prescale = 1

- Seemless counting continues
- Up count period is TxPR+1
Continuous-Up/Down Counting Mode
(Used for Symmetric PWM Waveforms)

This example:
TxCON.3-2 = 01 (reload TxCMP on underflow or period match)
TxPR = 3
TxCMP = 1 (initially)
Prescale = 1

- Seemless up/down repetition
- Up/down count period is 2*TxPR

![Diagram of Continuous-Up/Down Counting Mode](image)

- TxCMP loads with a 1
- TxCMP loads with a 2
- TxCMP loads with a 1

TxCNT Reg.
TxPWM/TxCMP (active high)
CPUCLK
PWM Outputs and Interrupts

- CPU Changes Period Reg. Buffer anytime here
- New Period is Auto-loaded on Underflow here

Timer Counter Value
- Comp2
- Comp1

TxCMP/TxPWM (active high)
- TxCMP/TxPWM (active low)

Compare Ints
Period Ints
Underflow Ints
GP Timer Registers

- As was the case with the period register, buffering is present for each timer compare register.

- Software writes a value to the compare register buffer, from which the TxCMP register is automatically loaded on one of three user selected events:
  - 1. timer underflow (TxCNT = 0)
  - 2. timer underflow or period match
  - 3. immediately

- The event selection is made using bits 2 and 3 of the TxCON register, and allows for on-the-fly compare value changes.
# GP Timer Registers

<table>
<thead>
<tr>
<th>Register</th>
<th>Address</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPTCONA</td>
<td>7400h</td>
<td>General Purpose Timer Control Register A</td>
</tr>
<tr>
<td>T1CNT</td>
<td>7401h</td>
<td>GP Timer 1 Counter Register</td>
</tr>
<tr>
<td>T1CMPR</td>
<td>7402h</td>
<td>GP Timer 1 Compare Register Buffer</td>
</tr>
<tr>
<td>T1PR</td>
<td>7403h</td>
<td>GP Timer 1 Period Register Buffer</td>
</tr>
<tr>
<td>T1CON</td>
<td>7404h</td>
<td>GP Timer 1 Control Register</td>
</tr>
<tr>
<td>T2CNT</td>
<td>7405h</td>
<td>GP Timer 2 Counter Register</td>
</tr>
<tr>
<td>T2CMPR</td>
<td>7406h</td>
<td>GP Timer 2 Compare Register Buffer</td>
</tr>
<tr>
<td>T2PR</td>
<td>7407h</td>
<td>GP Timer 2 Period Register Buffer</td>
</tr>
<tr>
<td>T2CON</td>
<td>7408h</td>
<td>GP Timer 2 Control Register</td>
</tr>
<tr>
<td>GPTCONB</td>
<td>7500h</td>
<td>General Purpose Timer Control Register B</td>
</tr>
<tr>
<td>T3CNT</td>
<td>7501h</td>
<td>GP Timer 3 Counter Register</td>
</tr>
<tr>
<td>T3CMPR</td>
<td>7502h</td>
<td>GP Timer 3 Compare Register Buffer</td>
</tr>
<tr>
<td>T3PR</td>
<td>7503h</td>
<td>GP Timer 3 Period Register Buffer</td>
</tr>
<tr>
<td>T3CON</td>
<td>7504h</td>
<td>GP Timer 3 Control Register</td>
</tr>
<tr>
<td>T4CNT</td>
<td>7505h</td>
<td>GP Timer 4 Counter Register</td>
</tr>
<tr>
<td>T4CMPR</td>
<td>7506h</td>
<td>GP Timer 4 Compare Register Buffer</td>
</tr>
<tr>
<td>T4PR</td>
<td>7507h</td>
<td>GP Timer 4 Period Register Buffer</td>
</tr>
<tr>
<td>T4CON</td>
<td>7508h</td>
<td>GP Timer 4 Control Register</td>
</tr>
</tbody>
</table>
GPTCONA Register (EVA)
GPTCONA @ 7400h

**GP Timer Status (read-only)**
- 0 = counting down
- 1 = counting up

**ADC start by event of GP Timer x**
- 00: no event starts ADC
- 01: setting of underflow interrupt flag
- 10: setting of period interrupt flag
- 11: setting of compare interrupt

**Compare Output Enable**
- 0 = all disabled (hi-impedanced)
- 1 = all enabled

**TxPWM/TxCMP Output Pin Conditioning**
- 00: forced low
- 01: active low
- 10: active high
- 11: forced high
**TxCON Register (EVA)**

**T1CON @ 7404h / T2CON @ 7408h**

### Upper Byte:

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>FREE</td>
</tr>
<tr>
<td>14</td>
<td>SOFT</td>
</tr>
<tr>
<td>13</td>
<td>reserved</td>
</tr>
<tr>
<td>12</td>
<td>TMODE1</td>
</tr>
<tr>
<td>11</td>
<td>TMODE0</td>
</tr>
<tr>
<td>10</td>
<td>TPS2</td>
</tr>
<tr>
<td>9</td>
<td>TPS1</td>
</tr>
<tr>
<td>8</td>
<td>TPS0</td>
</tr>
</tbody>
</table>

#### Emulation Halt Behavior
- 00 = stop immediately
- 01 = stop at end of period
- 1x = free run (do not stop)

#### Timer Clock Prescale
- 000: + 1
- 001: + 2
- 010: + 4
- 011: + 8
- 100: + 16
- 101: + 32
- 110: + 64
- 111: + 128

#### Count Mode Select
- 00 = stop/hold
- 01 = continuous-up/down
- 10 = continuous-up
- 11 = directional-up/down
PWM Review

- PWM is a scheme to represent a signal as a sequence of pulses
  - fixed carrier frequency
  - fixed pulse amplitude
  - pulse width proportional to instantaneous signal amplitude
  - PWM energy = original signal energy

- Differs from PAM (Pulse Amplitude Mod.)
  - fixed width,
  - variable amplitude
PWM Signal Representation

Original Signal

same areas (energy)

PWM representation

PAM representation
Why Use PWM in Digital Motor Control?

- Desired motor phase currents or voltages are known
- Power switching devices are transistors
  - Difficult to control in proportional region
  - Easy to control in saturated region
- PWM is a digital signal $\Rightarrow$ easy for DSP to output

![Diagram showing PWM control of a motor phase](image)
Asymmetric PWM Waveform

\[ T_{\text{PWM}} \]

Period

Compare

Counter

\[ T_{\text{pwm}} / T_{\text{cmp}} \text{ Pin} \]

(active high)

Caused by Period match
(toggle output in Asym mode only)

Caused by Compare match
Symmetric PWM Waveform
Compare Units

- There are three compare units. Each compare unit has two associated PWM outputs.

- They have capabilities beyond the GP timer compares, and feature programmable hardware deadband.

- The time base for the compare units is provided by GP timer 1.
## Compare Unit Registers

<table>
<thead>
<tr>
<th>Register</th>
<th>Address</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMCONA</td>
<td>7411h</td>
<td>Compare Control Register A</td>
</tr>
<tr>
<td>ACTRA</td>
<td>7413h</td>
<td>Compare Action Control Register A</td>
</tr>
<tr>
<td>DBTCONA</td>
<td>7415h</td>
<td>Dead-Band Timer Control Register A</td>
</tr>
<tr>
<td>CMPR1</td>
<td>7417h</td>
<td>Compare Register 1</td>
</tr>
<tr>
<td>CMPR2</td>
<td>7418h</td>
<td>Compare Register 2</td>
</tr>
<tr>
<td>CMPR3</td>
<td>7419h</td>
<td>Compare Register 3</td>
</tr>
<tr>
<td>COMCONB</td>
<td>7511h</td>
<td>Compare Control Register B</td>
</tr>
<tr>
<td>ACTRB</td>
<td>7513h</td>
<td>Compare Action Control Register B</td>
</tr>
<tr>
<td>DBTCONB</td>
<td>7515h</td>
<td>Dead-Band Timer Control Register B</td>
</tr>
<tr>
<td>CMPR4</td>
<td>7517h</td>
<td>Compare Register 4</td>
</tr>
<tr>
<td>CMPR5</td>
<td>7518h</td>
<td>Compare Register 5</td>
</tr>
<tr>
<td>CMPR6</td>
<td>7519h</td>
<td>Compare Register 6</td>
</tr>
</tbody>
</table>
Motivation for Dead-Band

- Dead-band control provides a convenient means of combating current shoot-through problems in a power converter.

- Shoot-through occurs when both the upper and lower gates in the same phase of a power converter are open simultaneously.

- This condition shorts the power supply and results in a large current draw.
Motivation for Dead-Band

- Shoot-through problems occur because transistors open faster than they close, and because high-side and low-side power converter gates are typically switched in a complimentary fashion.

- Although the duration of the shoot-through current path is finite during PWM cycling, (i.e. the closing gate will eventually shut), even brief periods of a short circuit condition can produce excessive heating and over stress in the power converter and power supply.
Motivation for Dead-Band - overview

Gate Signals are Complimentary PWM

- Transistor gates turn on faster than they shut off
- Short circuit if both gates are on at the same time!
Capture Units

- There are three capture units, and each is associated with a capture input pin.

- Each capture unit can choose GP timer 1 or 2 as its time base.

- The value of GP timer 1 or 2 is captured and stored in the corresponding 2-level-deep FIFO stack when a specified transition is detected on a capture input pin.
Capture Units

- Capture units timestamp transitions on capture input pins
- Three capture units - each associated with a capture input pin
Some Uses for the Capture Units

- Synchronized ADC start with capture event
- Measure the time width of a pulse
- Low speed velocity estimation from incr. encoder:

Problem: At low speeds, calculation of speed based on a measured position change at fixed time intervals produces large estimate errors.

Alternative: Estimate the speed using a measured time interval at fixed position intervals.

\[
v_k \approx \frac{x_k - x_{k-1}}{\Delta t}
\]
## Capture Units Registers

<table>
<thead>
<tr>
<th>Register</th>
<th>Address</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAPCON</td>
<td>7420h</td>
<td>Capture Control Register</td>
</tr>
<tr>
<td>CAPFIFO</td>
<td>7422h</td>
<td>Capture FIFO Status Register</td>
</tr>
<tr>
<td>CAP1FIFO</td>
<td>7423h</td>
<td>Two-Level Deep FIFO 1 Stack</td>
</tr>
<tr>
<td>CAP2FIFO</td>
<td>7424h</td>
<td>Two-Level Deep FIFO 2 Stack</td>
</tr>
<tr>
<td>CAP3FIFO</td>
<td>7425h</td>
<td>Two-Level Deep FIFO 3 Stack</td>
</tr>
<tr>
<td>CAP1FBOT</td>
<td>7427h</td>
<td>Bottom Register of FIFO 1</td>
</tr>
<tr>
<td>CAP2FBOT</td>
<td>7428h</td>
<td>Bottom Register of FIFO 2</td>
</tr>
<tr>
<td>CAP3FBOT</td>
<td>7429h</td>
<td>Bottom Register of FIFO 3</td>
</tr>
</tbody>
</table>
Quadrature Encoder Pulse (QEP)

- The QEP circuit, when enabled, decodes and counts the quadrature encoded input pulses on pins CAP1/QEP0 and CAP2/QEP1.

- The QEP circuit can be used to interface with an optical encoder to get position and speed information from a rotating machine.

- When the QEP circuit is enabled, the capture function on CAP1 and CAP2 pins is disabled.

- The QEP time base is provided by GP timer 2.
What is an Incremental Quadrature Encoder?
A digital (angular) position sensor
Analog-to-Digital Converter

- The LF2407 has a successive approximation 10-bit analogue-to-digital converter on-chip with a built-in sample-and-hold circuit.
- A total of 16 analogue multiplexed input channels are available.
- The fastest conversion time is 500ns with a 30 MHz clock and a prescale of 1.
- The converter is also capable of up to sixteen consecutive conversions performed 500ns apart.
Analog-to-Digital Converter

- There are two operating modes, the cascaded mode and the dual sequencer mode.
- In cascaded mode the two sequencers are operated in a single chain of up to sixteen consecutive conversions.
  - Each step of this chain can be performed with another AD-input or with the same input.
  - This is done with the help of four registers for channel selection.
- In dual sequencer mode there are two independent sequencers with 8 states each.
Analog-to-Digital Converter-overview

- 10-bit ADC core with built-in Sample & Hold (S/H)
- Sixteen multiplexed analog inputs (8 on C2402)
- Fast conversion time (S/H + conversion) of 500ns
- Autosequencing capability - up to 16 capability autoconversions
  - Two independent 8-state sequencers
  - “Dual-sequencer mode”
  - “Cascaded mode”
- Sixteen individually addressable result registers
- Multiple trigger sources for start-of-conversion
- Flexible interrupt control
## Analog-to-Digital Converter Registers

<table>
<thead>
<tr>
<th>Register</th>
<th>Address</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADCTRL1</td>
<td>70A0h</td>
<td>ADC Control Register 1</td>
</tr>
<tr>
<td>ADCTRL2</td>
<td>70A1h</td>
<td>ADC Control Register 2</td>
</tr>
<tr>
<td>MAX_CONV</td>
<td>70A2h</td>
<td>Maximum Conversion Channels Register</td>
</tr>
<tr>
<td>CHSELESEQ1</td>
<td>70A3h</td>
<td>Channel Select Sequencing Control Register 1</td>
</tr>
<tr>
<td>CHSELESEQ2</td>
<td>70A4h</td>
<td>Channel Select Sequencing Control Register 2</td>
</tr>
<tr>
<td>CHSELESEQ3</td>
<td>70A5h</td>
<td>Channel Select Sequencing Control Register 3</td>
</tr>
<tr>
<td>CHSELESEQ4</td>
<td>70A6h</td>
<td>Channel Select Sequencing Control Register 4</td>
</tr>
<tr>
<td>AUTO_SEQ_SR</td>
<td>70A7h</td>
<td>Autosequence Status Register</td>
</tr>
<tr>
<td>RESULT0</td>
<td>70A8h</td>
<td>Conversion Result Buffer Register 0</td>
</tr>
<tr>
<td>RESULT1</td>
<td>70A9h</td>
<td>Conversion Result Buffer Register 1</td>
</tr>
<tr>
<td>RESULT2</td>
<td>70AAh</td>
<td>Conversion Result Buffer Register 2</td>
</tr>
<tr>
<td>RESULT14</td>
<td>70B6h</td>
<td>Conversion Result Buffer Register 14</td>
</tr>
<tr>
<td>RESULT15</td>
<td>70B7h</td>
<td>Conversion Result Buffer Register 15</td>
</tr>
<tr>
<td>CALIBRATION</td>
<td>70B8h</td>
<td>Calibration Result (next conversion correction)</td>
</tr>
</tbody>
</table>
ADC Control Register 1 - Upper Byte

ADCTRL1 @ 70A0h

ADC Module Reset
0 = no effect
1 = reset (set back to 0 by ADC logic)

Acquisition Time Prescale (S/H)
Value = (binary+1) x 2
* Time dependent on the “Conversion Clock Prescale” bit (Bit 7 “CPS”)

<table>
<thead>
<tr>
<th>15</th>
<th>14</th>
<th>13</th>
<th>12</th>
<th>11</th>
<th>10</th>
<th>9</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>reserved</td>
<td>RESET</td>
<td>SOFT</td>
<td>FREE</td>
<td>ACQ_PS3</td>
<td>ACQ_PS2</td>
<td>ACQ_PS1</td>
<td>ACQ_PS0</td>
</tr>
</tbody>
</table>

Emulation Halt Behavior
00 = stop immediately
10 = stop after current conversion
x1 = free run (do not stop)
ADC Control Register 1 - Lower Byte
ADCTRL1 @ 70A0h

Continuous Run
0 = stops after reaching end of sequence
1 = continuous (starts all over again from “initial state”)

Sequencer Mode
0 = dual mode
1 = cascaded mode

Conversion Prescale
0 = CLK / 1
1 = CLK / 2

Interrupt Priority
0 = high
1 = low

Calibration and Self-Test Functions
ADC Control Register 2 - Upper Byte
ADCTRL1 @ 70A1h

- **EVB SOC**
  - (cascaded mode only)
  - 0 = no action
  - 1 = start by EVB

- **Interrupt Flag (SEQ1)**
  - 0 = no interrupt
  - 1 = interrupt occurred

- **Start Conversion (SEQ1)**
  - 0 = clear pending SOC trigger
  - 1 = software trigger-start SEQ1

- **EVA SOC**
  - SEQ1 Mask Bit
  - 0 = cannot be started by EVA trigger
  - 1 = can be started by EVA trigger

- **Reset SEQ1 / Start Calibration**
  - 0 = no action
  - 1 = immediate reset or start calibration if bit 3 of ADCTRL1=1

- **SEQ1 Busy**
  - 0 = idle
  - 1 = in progress

- **Interrupt Mode Enable (SEQ1)**
  - 00 = interrupt disabled
  - 01 = on INT_FLAG_SEQ1 set
  - 10 = interrupt every other EOS
  - 11 = reserved
ADC Control Register 2 - Lower Byte

ADCTRL1 @ 70A1h

- **External SOC (SEQ1)**
  - 0 = no action
  - 1 = start by signal from ADCSOC pin

- **Interrupt Flag (SEQ2)**
  - 0 = no interrupt
  - 1 = interrupt occurred

- **Start Conversion (SEQ2)**
  - (dual-sequencer mode only)
  - 0 = clear pending SOC trigger
  - 1 = software trigger-start SEQ2

- **SEV SOC**
  - 0 = cannot be started by EVB trigger
  - 1 = can be started by EVB trigger

- **Reset SEQ2**
  - 0 = no action
  - 1 = immediate reset SEQ2 to “initial state”

- **SEQ2 Busy**
  - 0 = idle
  - 1 = in progress

- **Interrupt Mode Enable (SEQ2)**
  - 00 = interrupt disabled
  - 01 = on INT_FLAG_SEQ2 set
  - 10 = interrupt every other EOS
  - 11 = reserved

- **SEQ2 Mask bit**
  - 0 = cannot be started by EVB trigger
  - 1 = can be started by EVB trigger
Maximum Conversion Channels Register

MAX_CONV @ 70A2h

- Bit fields define the maximum number of autoconversions (binary+1)

Dual Mode

SEQ2

SEQ1

- Autoconversion session always starts with the "initial state" and continues sequentially until the "end state", if allowed

<table>
<thead>
<tr>
<th></th>
<th>SEQ1</th>
<th>SEQ2</th>
<th>Cascaded</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial state</td>
<td>CONV00</td>
<td>CONV08</td>
<td>CONV00</td>
</tr>
<tr>
<td>End state</td>
<td>CONV07</td>
<td>CONV15</td>
<td>CONV15</td>
</tr>
</tbody>
</table>
ADC Input Channel Select Sequencing Control Register

<table>
<thead>
<tr>
<th>Bits 15-12</th>
<th>Bits 11-8</th>
<th>Bits 7-4</th>
<th>Bits 3-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>70A3h</td>
<td>CONV03</td>
<td>CONV02</td>
<td>CONV01</td>
</tr>
<tr>
<td>70A4h</td>
<td>CONV07</td>
<td>CONV06</td>
<td>CONV05</td>
</tr>
<tr>
<td>70A5h</td>
<td>CONV11</td>
<td>CONV10</td>
<td>CONV09</td>
</tr>
<tr>
<td>70A6h</td>
<td>CONV15</td>
<td>CONV14</td>
<td>CONV13</td>
</tr>
</tbody>
</table>

CHSELSEQ1
CHSELSEQ2
CHSELSEQ3
CHSELSEQ4
ADC Conversion Result Buffer Register
RESULT0 @ 70A8h through RESULT15 @ 70B7h (Total of 16 Registers)

With \( V_{\text{REFHI}} = 3.3 \, \text{V} \), and \( V_{\text{REFLO}} = 0 \, \text{V} \), we have:

<table>
<thead>
<tr>
<th>analog volts</th>
<th>converted value</th>
<th>RESULTx</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3</td>
<td>3FFh</td>
<td>1111111110000000</td>
</tr>
<tr>
<td>1.65</td>
<td>1ffh</td>
<td>0111111110000000</td>
</tr>
<tr>
<td>0.00322</td>
<td>1h</td>
<td>0000000100000000</td>
</tr>
<tr>
<td>0</td>
<td>0h</td>
<td>0000000000000000</td>
</tr>
</tbody>
</table>
Several methods of implementing a TMS320LF2407 communications system are possible. The method selected for a particular design should reflect the method that meets the required data rate at the lowest cost.

- Serial Peripheral Interface (SPI)
- Serial Communication Interface (SCI)
- Controller Area Network (CAN)
TMSLF2407 DSP CONTROLLER

Review
TMS320C240x Block Diagram
Multiplier and ALU / Shifters
Program Memory

![Diagram of Program Memory]

- **Program Bus**
- **Data Bus**
- **Address**
  - Program ROM / FLASH Instruction
- **STACK (8x16)**
- **MUX**
- **PC**
- **To Data Memory**

SUN Dan
College of Electrical Engineering, Zhejiang University

2008-9-18
Data Memory
# TMS320C240x Memory Map

<table>
<thead>
<tr>
<th>Hex</th>
<th>Program</th>
<th>Hex</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>Interrupts</td>
<td>0000</td>
<td>Memory-Mapped Registers</td>
</tr>
<tr>
<td>0040</td>
<td>On-chip ROM / Flash (External if MP/MC = 1)</td>
<td>0060</td>
<td>On-chip DARAM B2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0080</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0200</td>
<td>On-chip DARAM B0 (CNF = 0) or Reserved (CNF = 1)</td>
</tr>
<tr>
<td>X</td>
<td>External</td>
<td>0300</td>
<td>On-chip DARAM B1</td>
</tr>
<tr>
<td>8000</td>
<td>SARAM (2K) (PON = 1) or External (PON = 0)</td>
<td>0400</td>
<td>Reserved</td>
</tr>
<tr>
<td>8800</td>
<td>External</td>
<td>0800</td>
<td>SARAM (2K) (DON = 1) or Reserved (DON = 0)</td>
</tr>
<tr>
<td>FE00</td>
<td>Reserved (CNF = 1)</td>
<td>7000</td>
<td>Non-EV Peripherals</td>
</tr>
<tr>
<td></td>
<td>External (CNF = 0)</td>
<td>7400</td>
<td>EV Peripherals</td>
</tr>
<tr>
<td>FF00</td>
<td>On-chip DARAM B0 (CNF = 1) or External (CNF = 0)</td>
<td>7540</td>
<td>Reserved</td>
</tr>
<tr>
<td>FFFF</td>
<td></td>
<td>8000</td>
<td>External</td>
</tr>
</tbody>
</table>
## Pipeline Process

<table>
<thead>
<tr>
<th>Cycle</th>
<th>100</th>
<th>101</th>
<th>102</th>
<th>103</th>
<th>104</th>
<th>105</th>
<th>106</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>F₁</td>
<td>D₁</td>
<td>R₁</td>
<td>E₁</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sub</td>
<td>F₂</td>
<td>D₂</td>
<td>R₂</td>
<td>E₂</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mpy</td>
<td>F₃</td>
<td>D₃</td>
<td>R₃</td>
<td>E₃</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Store</td>
<td>F₄</td>
<td>D₄</td>
<td>R₄</td>
<td>E₄</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Fully loaded pipeline (normal operation)*
Peripherals
TMS320C240x Instruction Set

Accumulator Memory Reference
- ABS, NEG, SUB
- ADD, NORM, SUBB
- ADDC, OR, SUBC
- ADDS, ROL, SUBS
- ADDT, ROR, SUBT
- AND, SACH, XOR
- CMPL, SAACL, ZALR

Auxiliary Register and Data Page Pointer
- ADRK, MAR
- CMPR, SAR
- LAR, SBRK
- LDP

Control Instructions
- BIT, POP
- BITT, POPD
- CLRC, PSHD
- IDLE, PUSH
- RPT, SETC
- LST, SST
- NOP

I/O and Data Memory Operations
- BLDD, OUT
- BLPD, SPLK
- DMOV, TBLR
- IN, TBLW

Multiply, T, P
- APAC, MPYA
- LPH, MPYS
- LT, MPYU
- LTA, PAC
- LTD, SPAC
- LTP, SPH
- LTS, SPL
- MAC, SPM
- MACD, SQRA
- MPY, SQRS

Branch
- B, CC
- BACC, INTR
- BANZ, NMI
- BCND, RET
- CALA, RETC
- CALL, TRAP

SUN Dan
College of Electrical Engineering, Zhejiang University
The end