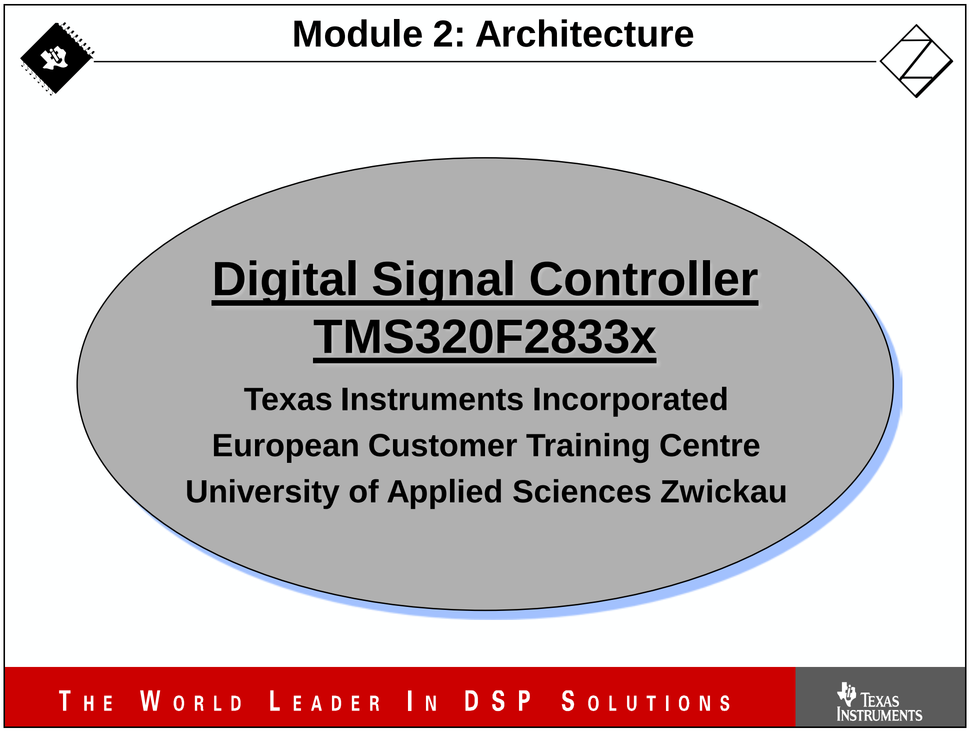
Architecture

Introduction

The TMS320F2833x Digital Signal Controller is capable of executing six basic operations in a single instruction cycle, and therefore the architecture of the device must reflect this feature in some way. Remember this key point when we look into the details of this Digital Signal Controller (DSC). It will help you to understand the ‘philosophy’ behind the device with its different hardware units. Doing six basic maths operations is no magic; we will find all the hardware modules that are required to do so in this chapter.

In this and other modules, we will discuss the following parts of the architecture:

- Internal bus structure
- CPU
- Direct Memory Access Controller
- Floating-point Arithmetic Unit
- Fixed-point Hardware Multiplier, Arithmetic-Logic-Unit, Hardware-Shifter
- Pipeline Processing of Instructions
- Memory Map



Module 2: Architecture

Digital Signal Controller
TMS320F2833x

Texas Instruments Incorporated
European Customer Training Centre
University of Applied Sciences Zwickau

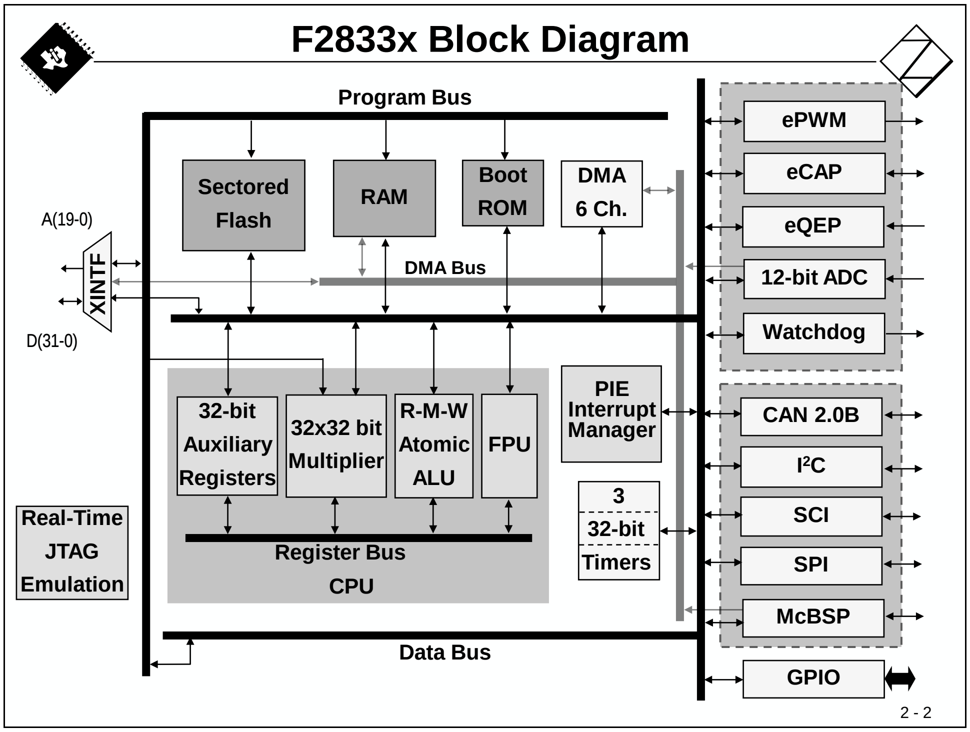
THE WORLD LEADER IN DSP SOLUTIONS

Module Topics

TMS320F2833x Block Diagram

The TMS320F2833x Block Diagram can be divided into the following functional units:

- Internal and external Bus System
- Central Processing Unit (CPU)
- Internal Memory Sections
- Control Peripherals
- Communication Channels
- Direct Memory Access Controller (DMA)
- Interrupt Management Unit (PIE) and Core Time Unit
- Real - Time Emulation Interface


F2833x Block Diagram

Bus System

Since the core of the TMS2833x Microcontroller is a DSP, it must be able to read at least two operands from memory and transfer them to the central processing unit in a single clock cycle. To do so, the F2833x features two independent bus systems, called the "Program Bus" and the "Data Bus". This type of processor technology is called “Harvard-Architecture”. Due to the ability of the F2833x to read operands not only from data memory but also from program memory, Texas Instruments calls its technology a “modified Harvard-Architecture”. The “bypass”-arrow in the bottom left corner of Slide 2-2 indicates this additional feature.

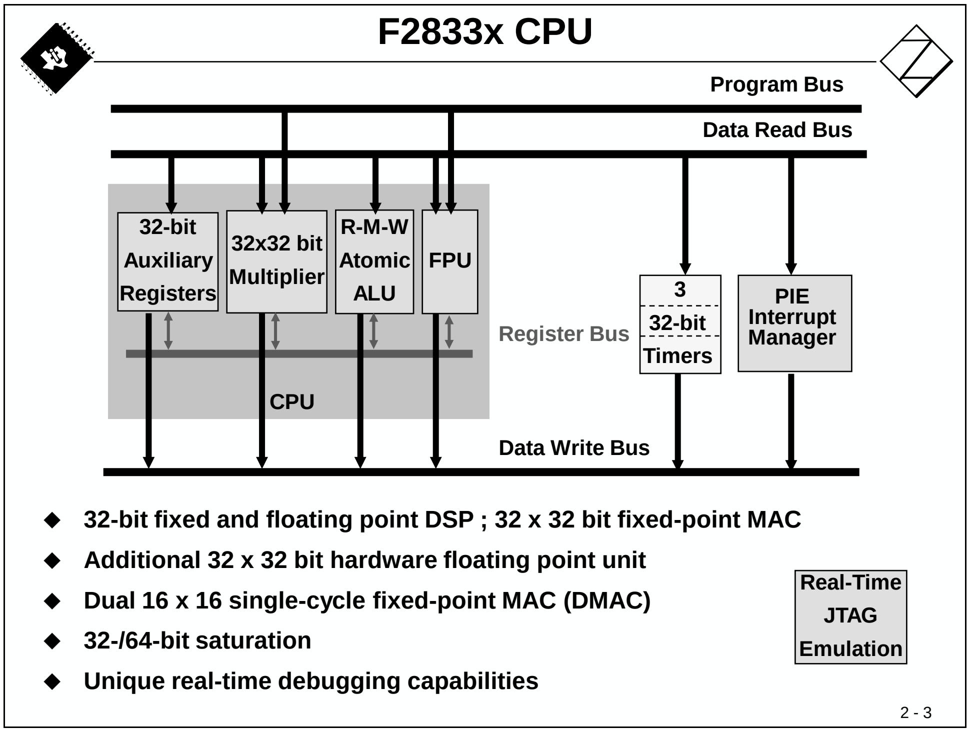
In addition, the F2833x connects all units inside the CPU core to a third bus system, called the “Register Bus”, allowing a very fast exchange of data between its parallel mathematical units. Finally, because the DMA unit is able to operate on certain parts of the hardware units independently of the CPU, a "Direct Memory Access Bus" has been added for this purpose.

On the left hand side of slide 2-2 you will notice a multiplexer block for data (D31-D0) and address (A19-A0). This is an interface to connect external devices to the F2833x. Please note that you cannot access the external program bus data and the data bus data simultaneously. Compared to a single cycle for internal access to two 32-bit operands, it takes at least 2 cycles to do the same with external memory, not taking into account additional wait cycles for slower external memories!

Central Processing Unit (CPU)

The F2833x - CPU is able to execute most of the instructions to perform register-to-register operations and a range of instructions that are commonly used by microcontrollers, e.g. byte packing and unpacking and bit manipulation in a single cycle. The architecture is also supported by powerful addressing modes, which allow the compiler as well as the assembly programmer to generate compact code that almost corresponds one-to-one with the C code.

The F2833x is as efficient in typical math tasks for Digital Signal Processing as it is in the system control tasks that are typically handled by microcontroller devices. This efficiency removes the need for a second processor in many systems.



F2833x CPU

- 32-bit fixed and floating point DSP ; 32 x 32 bit fixed-point MAC
- Additional 32 x 32 bit hardware floating point unit
- Dual 16 x 16 single-cycle fixed-point MAC (DMAC)
- 32-/64-bit saturation
- Unique real-time debugging capabilities

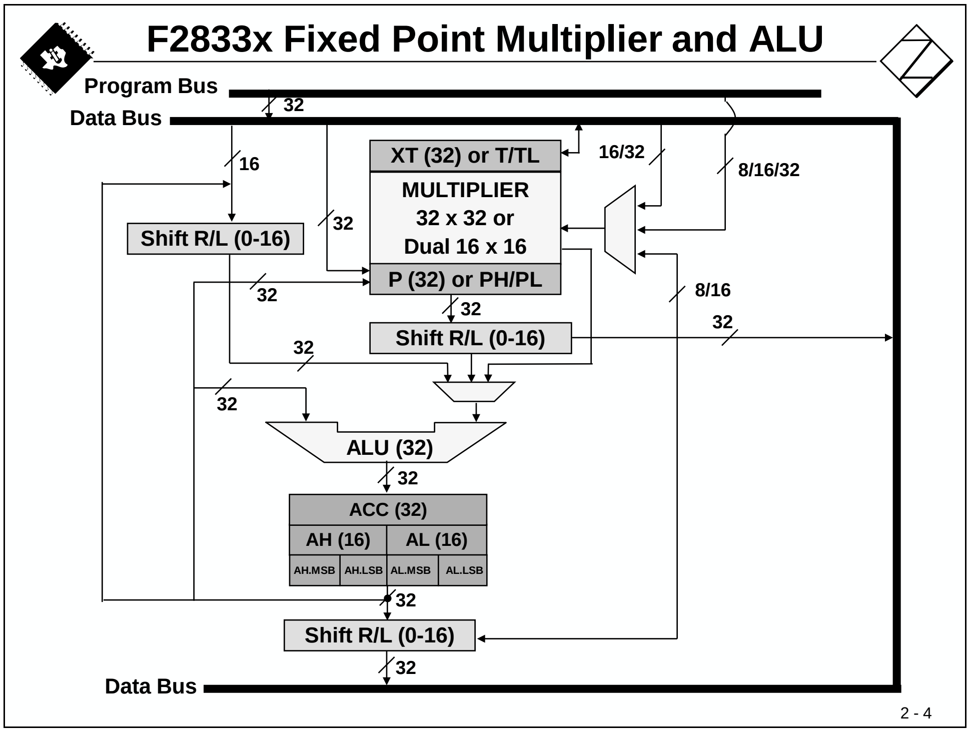
Three 32-bit timers can be used for general timing purposes or to generate hardware driven time periods for real-time operating systems. The Peripheral Interrupt Expansion Manager (PIE) allows fast interrupt response to the various sources of external and internal signals and events. The PIE-Manager processes individual interrupt vectors for all sources and reduces the response time to an external event, called "Interrupt Latency", to an absolute minimum.

A fixed-point 32-bit by 32-bit hardware multiplier and a 32-bit arithmetic logic unit (ALU) can be used in parallel to simultaneously execute a multiply and an addition operation on fixed-point numbers. The auxiliary register group is equipped with its own arithmetic unit (ARAU)-also used in parallel to perform pointer arithmetic. In addition, a hardware floating-point unit (FPU) for IEEE-754 single point precision numbers allows the direct usage of floating-point numbers from C or MatLab-code.

The JTAG-interface is a very powerful tool to support real-time data exchange between the DSC and a host during the debug phase of project development. A special operating mode called "Real-time Debug" allows variables to be monitored while the code is running in real-time, without a single clock cycle delay to the control code.

Fixed-point Math Unit

The 32-bit by 32-bit fixed-point “Multiply and Accumulate (MAC)” capabilities of the F2833x and its internal 64-bit processing capabilities enable this DSC to efficiently handle higher numerical resolution problems based on fixed-point numbers. In control applications, a lot of algorithms are numerically based on a technique, called "Binary Fractions", sometimes also called "Integer-Quotient (IQ)"-numbers. The F2833x directly supports this approach. Furthermore, if the samples and coefficients are stored in a binary number of 16 bits or less, which is quite often possible, this controller is able to perform two 16-bit by 16-bit multiply and accumulate instructions simultaneously. The terminology for such an instruction is “Dual Multiply and Accumulate (DMAC)”.



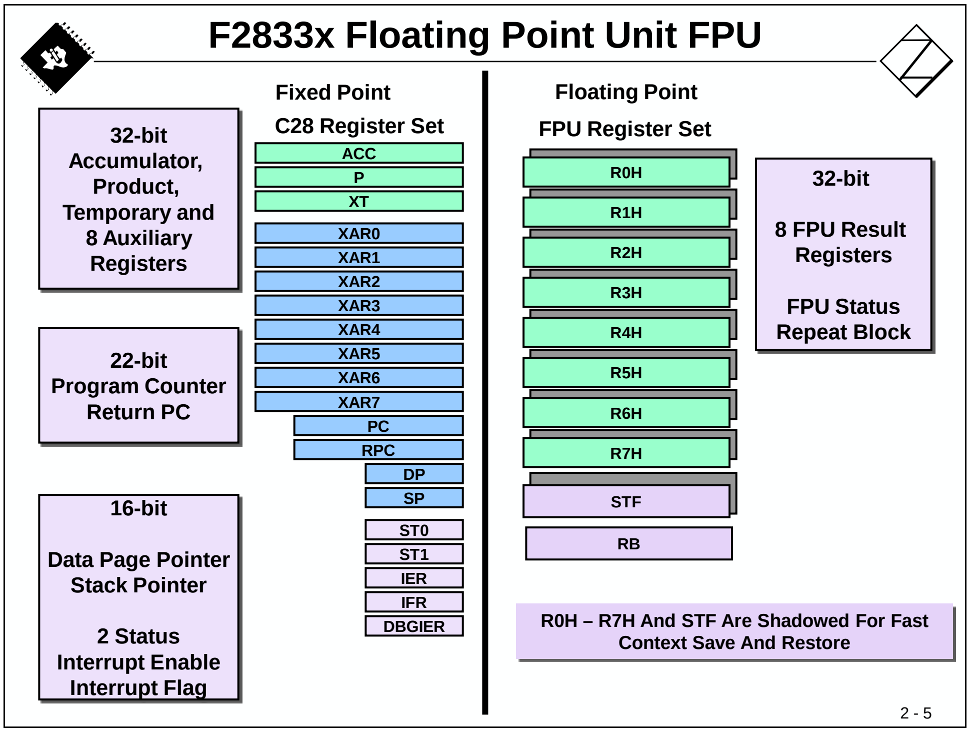
Fixed-point multiplication uses the XT ("eXtended Temp") register to hold the first operand and multiply it by a second operand, which is loaded from memory. If XT is loaded from a data memory location and the second operand is fetched from a program memory location, a single-cycle multiply operation can be performed. The result of a multiplication is shifted into the P ("Product") register or directly into the Accumulator (ACC). Remember, if you multiply a 32-bit by a 32-bit number, what size is the result? Answer: 64-bits. The F2833x instruction set includes two groups of multiply operations to store both 32-bit portions of the result into P and ACC. In this way, we can say that the registers ACC and P are combined to form a single 64-bit register.

Three hardware shifters can be used in parallel with other hardware units of the CPU. Shifters are usually used to scale intermediate results in a real-time control loop or just to multiply/divide by numbers of type 2^n.

The Arithmetic Logic Unit (ALU) performs all other mathematical operations other than multiplication. The first operand is always the content of the Accumulator (ACC) or a part of it. The second operand for an operation is loaded from data memory, from program memory, from the P register or directly from the multiply unit.

Floating-point Math Unit

To add more flexibility to the device by using single precision floating-point data types in C, Texas Instruments have added a second hardware multiply unit to the F2833x family. The right hand side of Slide 2-5 shows all the floating-point registers. The 8 general purpose registers (R0H to R7H) are supported by a status register (STF) and a repeat block register (RB). The latter is used to execute a block of machine code without the need for a software loop. Such a technique allows the pipeline of the CPU to run at faster speeds.



F2833x Floating Point Unit FPU

Fixed Point

C28 Register Set

32-bit Accumulator, Product, Temporary and 8 Auxiliary Registers

ACC
P
XT
XAR0
XAR1
XAR2
XAR3
XAR4
XAR5
XAR6
XAR7

22-bit Program Counter Return PC

PC
RPC

16-bit

Data Page Pointer
Stack Pointer

2 Status
Interrupt Enable
Interrupt Flag

DP
SP
ST0
ST1
IER
IFR
DBGIER

Floating Point

FPU Register Set

R0H
R1H
R2H
R3H
R4H
R5H
R6H
R7H
STF
RB

32-bit

8 FPU Result Registers

FPU Status
Repeat Block

R0H – R7H And STF Are Shadowed For Fast Context Save And Restore

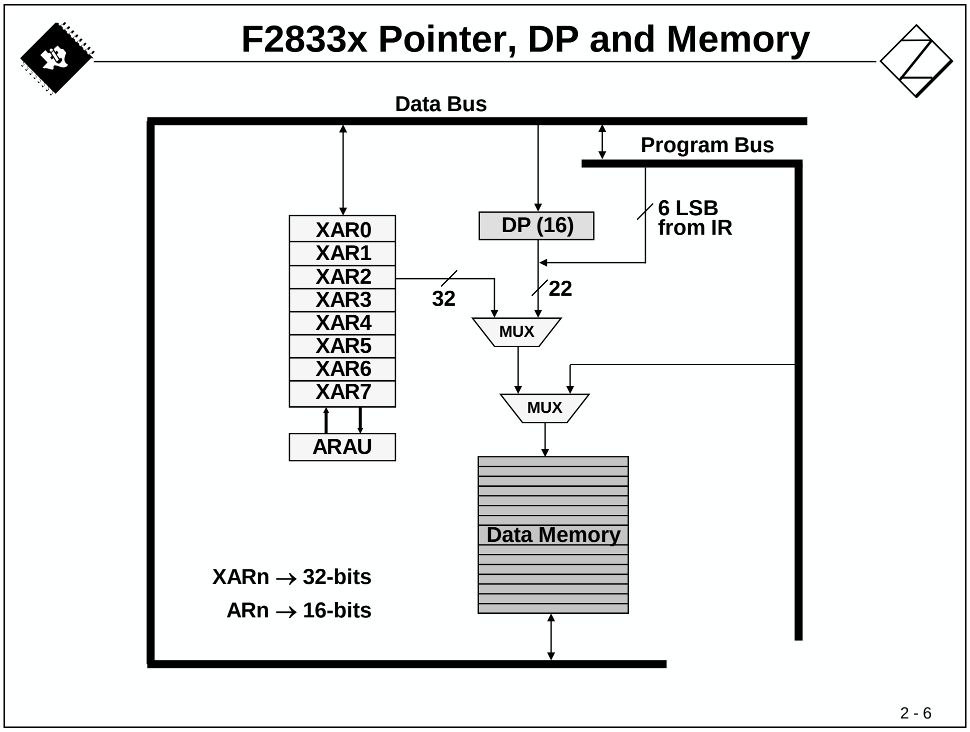
2 - 5

The left hand part of Slide 2-5 shows the fixed-point register set. It consists of the 3 CPU registers, Accumulator (ACC), Product (P) and extended temp (XT), 8 general purpose registers (XAR0…XAR7) and a set of control and status registers, such as "Program Counter" (PC), "Data Page Pointer"(DP), "Stack Pointer" (SP), "Interrupt Enable" (IER), "Interrupt Flag" (IFR) and "Debug Interrupt Enable" (DBGIER).

Data Memory Access

Two basic methods are available to access data memory locations:

- Direct Addressing Mode
- Indirect Addressing Mode



F2833x Pointer, DP and Memory

XARn → 32-bits

ARn → 16-bits

2 - 6

Direct addressing mode generates the 22-bit address for a memory access from two sources - a 16-bit register “Data Page (DP)” for the highest 16 bits plus another 6 bits taken from the instruction. Advantage: Once DP is set, we can access any location of the selected page, in any order. Disadvantage: If the code needs to access another page, DP must be changed first.

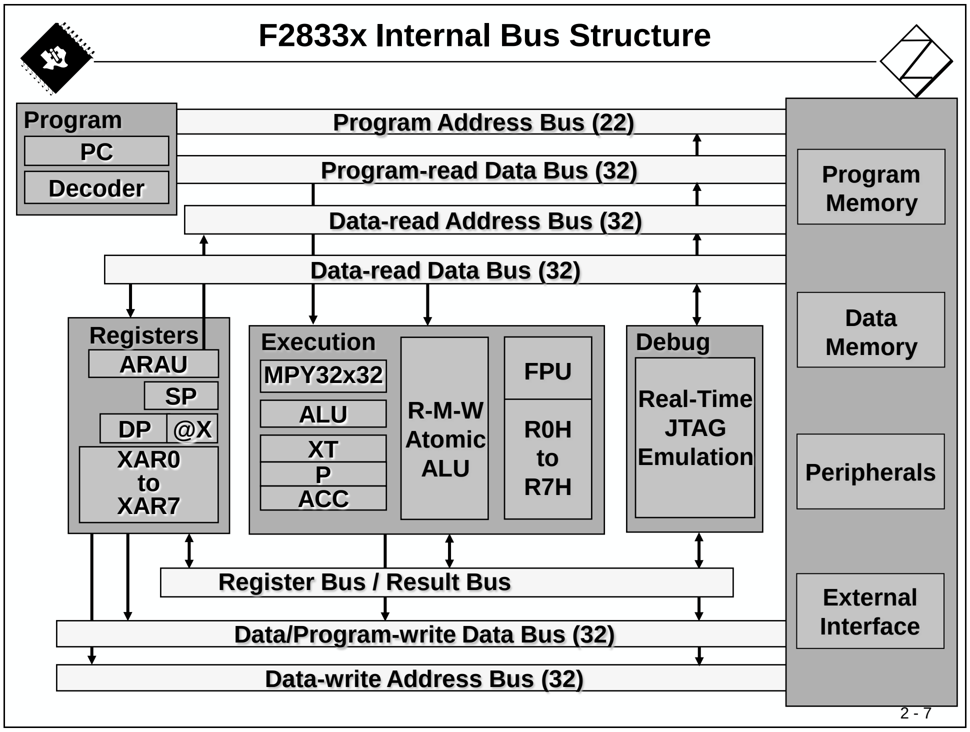
Indirect addressing mode uses one of eight 32-bit XARn registers to hold the 32-bit address of the operand. Advantage: With the help of the ARAU, pointer arithmetic is available in the same cycle in which an access to a data memory location is made. Disadvantage: A random access to data memory needs the pointer register to be setup with a new value.

The auxiliary register arithmetic unit (ARAU) is able to perform pointer manipulations in the same clock cycle as the access is made to a data memory location. The options for the ARAU are: post-increment, pre-decrement, index addition and subtraction, stack relative operation, circular addressing and bit-reverse addressing with additional options.

Internal Bus Structure

A typical feature of Digital Signal Processors is their ability to increase the data throughput based on multiple busses. Such busses are used to move data between memory locations, peripheral units and the CPU. The F2833x memory bus architecture contains:

- A program read bus (22-bit address line and 32-bit data line)
- A data read bus (32-bit address line and 32-bit data line)
- A data write bus (32-bit address line and 32-bit data line)
- A register bus (32-bit data line and direct register addressing)



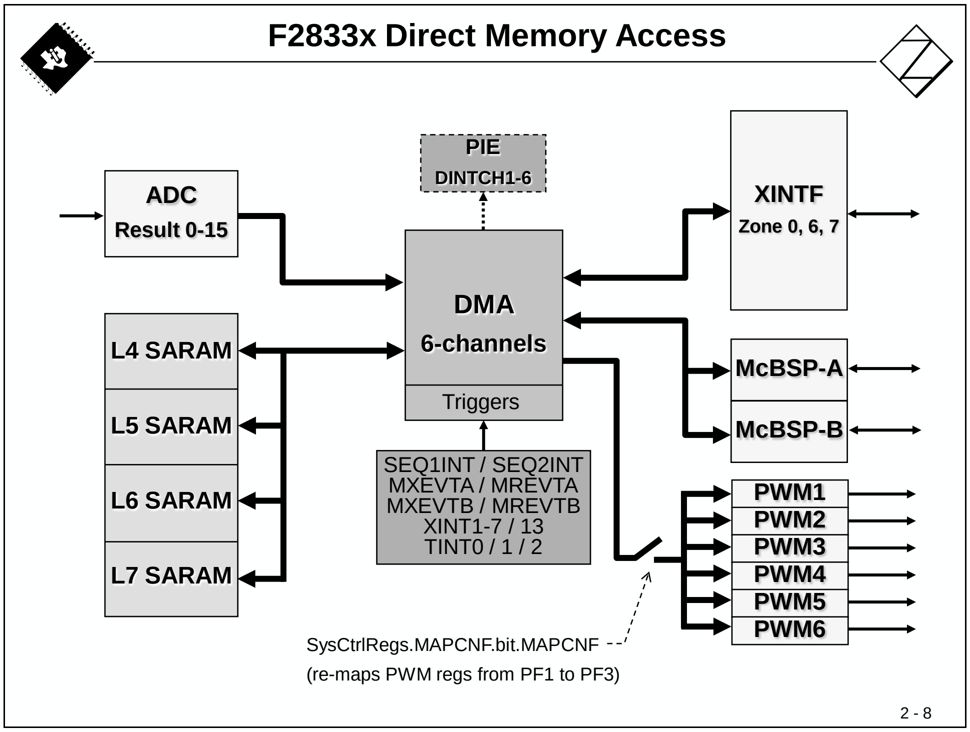
The 32-bit wide data busses allow single cycle 32-bit operations. This multiple bus architecture, known as a Harvard Bus Architecture enables the F2833x to (1) fetch an instruction, (2) read a first data value and (3) write a second data value all within in a single clock cycle.

All registers to control peripheral units are mapped into specific locations in data memory space and can be accessed with an ordinary data memory write or read instruction. For important peripheral registers, some security mechanisms are implemented to prevent a modification by accident.

All internal memory sections are attached both to program and data memory (called "unified memory model"). It allows the designer to select a certain part to be used as code or as a data section.

Direct Memory Access Controller (DMA)

A Direct Memory Access Controller has been introduced in the F2833x family. A DMA unit allows a data transfer from a source to a destination unit without the need of an interaction of the CPU. The strength of a digital signal controller (DSC) is not measured purely in processor speed, but also in total system capability. As a part of the equation, whenever the CPU bandwidth for a given function can be reduced, the greater the system capability will be. Very often applications spend a significant amount of their bandwidth moving data, whether it is from off-chip memory to on-chip memory or from a peripheral such as an analog-to-digital converter (ADC) to RAM, or even from one peripheral to another. Furthermore, there are times when this data comes in a format that is not compatible with the optimum processing powers of the CPU. The DMA module has the ability to free up CPU bandwidth and rearrange the data into a pattern for more streamlined processing.


F2833x Direct Memory Access

SysCtrlRegs.MAPCNF.bit.MAPCNF
(re-maps PWM regs from PF1 to PF3)

2 - 8

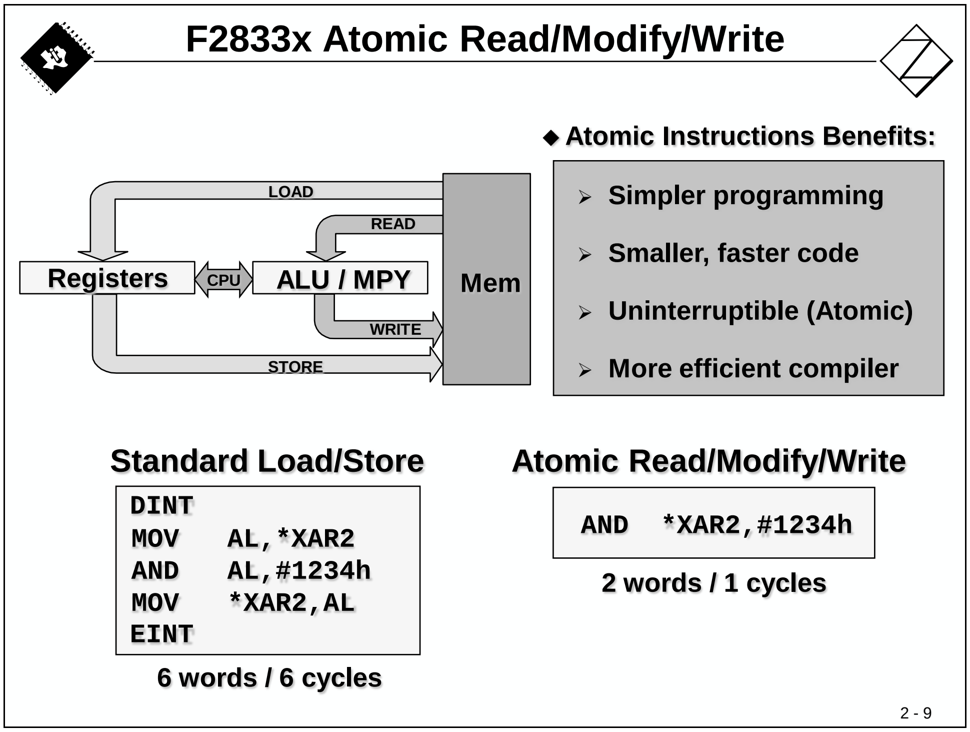
The DMA module is an event-based machine, meaning it requires a peripheral interrupt trigger to start a DMA transfer, such as:

- Analogue to Digital Converter Sequencer 1 (SEQ1INT) or Sequencer 2 (SEQ2INT)
- Multichannel Buffered Serial Port A and B (McBSP-A, McBSP-B) transmit/receive
- External Interrupt Input Signals XINT1-7 and XINT13
- CPU Timers 0, 1 and 2
- Pulse Width Module (PWM) signals ePWM1-6
- Software

As data sources and/or destinations that can be initialized:

- Internal SARAM sections L4 to L7
- All external memory zones XINTF
- ADC result registers (source only)
- McBSP-A and McBSP-B transmit and receive buffers
- PWM units 1-6 (destination only)

Atomic Arithmetic Logic Unit (ALU)



F2833x Atomic Read/Modify/Write

LOAD

READ

Registers

CPU

ALU / MPY

Mem

WRITE

STORE

- **Atomic Instructions Benefits:**
  - Simpler programming
  - Smaller, faster code
  - Uninterruptible (Atomic)
  - More efficient compiler

Standard Load/Store

```
DINT
MOV   AL,*XAR2
AND   AL,#1234h
MOV   *XAR2,AL
EINT
```

6 words / 6 cycles

Atomic Read/Modify/Write

```
AND   *XAR2,#1234h
```

2 words / 1 cycles

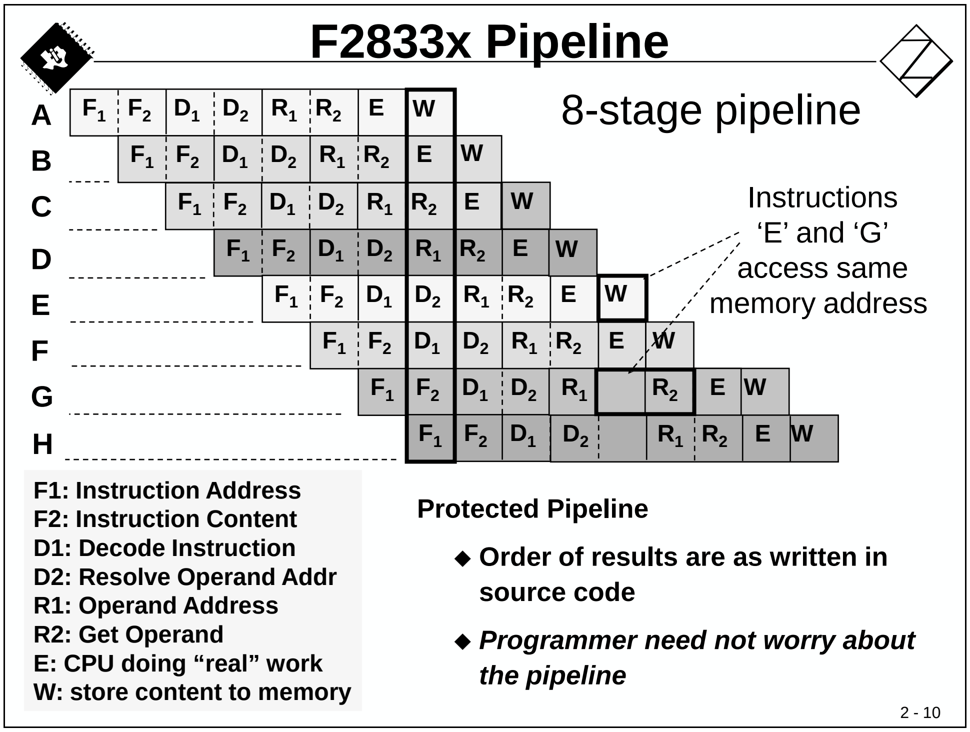
2 - 9

Atomic instructions are common with embedded system controllers. Examples are logical operations, such as AND, OR and EXOR directly performed in data memory locations. Usually, these instructions must be executed without an interruption between read and write accesses; they are called "non-interruptible" or "atomic" instructions. The F2833x atomic Arithmetic Logic Unit (ALU) capability supports such types of instructions; as shown on the right hand side of Slide 2-9.

By contrast, the traditional coding (left hand side of Slide 2-9) would execute several cycles slower than atomic instructions.

Instruction Pipeline

Like almost all today's microprocessors that operate in speed regions above 50 MHz the F2833x also uses a pipeline technique to maximize the code throughput. The F2833x features an 8-stage protected pipeline. The adjective "protected" means that the pipeline unit itself automatically prevents a "write to" and a "read from" the same location from occurring out of sequence (see instructions E and G in Slide 2-10). This pipelining also enables the F283xx to execute at high speeds without resorting to expensive high-speed memories. An additional branch-look-ahead hardware minimizes the delay when jumping to another address. Particular assembly instructions called "conditional store operations" avoid pipeline stalls and further improve the overall system performance.



F2833x Pipeline

8-stage pipeline

A	F_1	F_2	D_1	D_2	R_1	R_2	E	W								
B		F_1	F_2	D_1	D_2	R_1	R_2	E	W							
C			F_1	F_2	D_1	D_2	R_1	R_2	E	W						
D				F_1	F_2	D_1	D_2	R_1	R_2	E	W					
E					F_1	F_2	D_1	D_2	R_1	R_2	E	W				
F						F_1	F_2	D_1	D_2	R_1	R_2	E	W			
G							F_1	F_2	D_1	D_2	R_1		R_2	E	W	
H								F_1	F_2	D_1	D_2		R_1	R_2	E	W

Instructions 'E' and 'G' access same memory address

F1: Instruction Address
F2: Instruction Content
D1: Decode Instruction
D2: Resolve Operand Addr
R1: Operand Address
R2: Get Operand
E: CPU doing "real" work
W: store content to memory

Protected Pipeline

- **Order of results are as written in source code**
- ***Programmer need not worry about the pipeline***

2 - 10

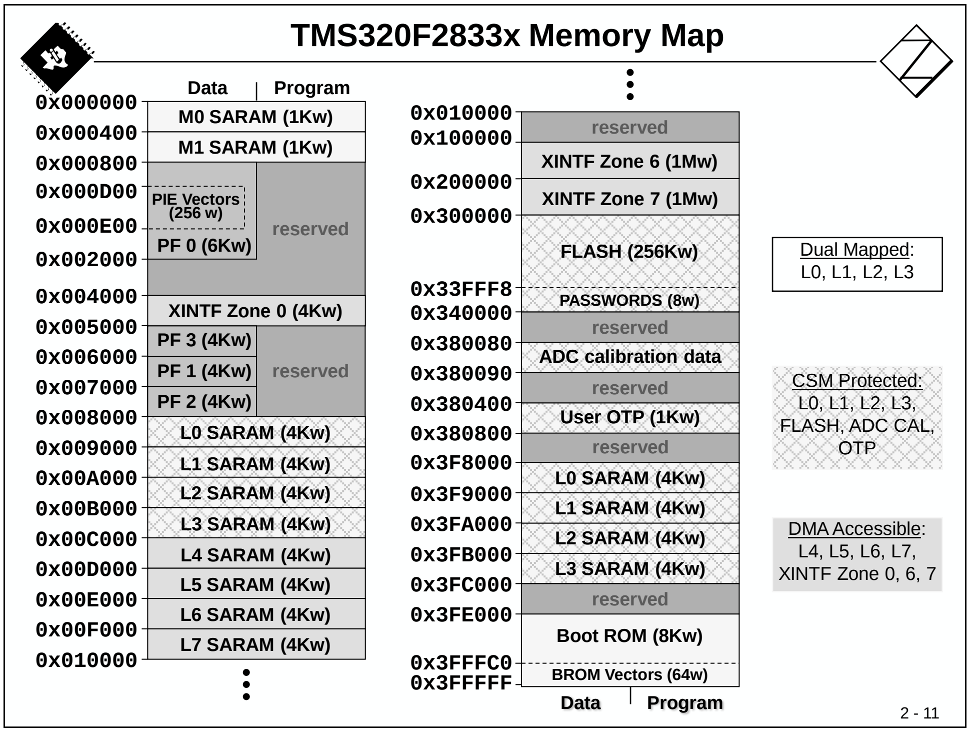
Each instruction passes through 8 stages until final completion. Once the pipeline is filled with instructions, one instruction is executed per clock cycle. For a 150MHz device, this equates to 6.67ns per instruction.

The stages are:

F1: Generate Instruction Address at program bus address lines.
F2: Read the instruction from program bus data lines.
D1: Decode Instruction
D2: Calculate Address information for operand(s) of the instruction
R1: Load operand(s) address to data and/or program bus address lines
R2: Read Operand
X: Execute the instruction
W: Write back result to data memory

Memory Map

The memory space of the F2833x is divided into program space and data space. There are several different types of memory available that can be used as both as a program or a data space member. These include independent sections of flash memory, single access RAM (SARAM), one time programmable memory (OTP) and boot ROM. The latter is factory programmed with boot software routines and trigonometric lookup tables used in maths based algorithms. Memory space width is always 16 bits.



TMS320F2833x Memory Map

	Data | Program	
0x000000	M0 SARAM (1Kw)	
0x000400	M1 SARAM (1Kw)	
0x000800		reserved
0x000D00	PIE Vectors (256 w)	
0x000E00	PF 0 (6Kw)	
0x002000		
0x004000	XINTF Zone 0 (4Kw)	
0x005000	PF 3 (4Kw)	reserved
0x006000	PF 1 (4Kw)	
0x007000	PF 2 (4Kw)	
0x008000	L0 SARAM (4Kw)	
0x009000	L1 SARAM (4Kw)	
0x00A000	L2 SARAM (4Kw)	
0x00B000	L3 SARAM (4Kw)	
0x00C000	L4 SARAM (4Kw)	
0x00D000	L5 SARAM (4Kw)	
0x00E000	L6 SARAM (4Kw)	
0x00F000	L7 SARAM (4Kw)	
0x010000		

0x010000	reserved
0x100000	XINTF Zone 6 (1Mw)
0x200000	XINTF Zone 7 (1Mw)
0x300000	FLASH (256Kw)
0x33FFF8	PASSWORDS (8w)
0x340000	reserved
0x380080	ADC calibration data
0x380090	reserved
0x380400	User OTP (1Kw)
0x380800	reserved
0x3F8000	L0 SARAM (4Kw)
0x3F9000	L1 SARAM (4Kw)
0x3FA000	L2 SARAM (4Kw)
0x3FB000	L3 SARAM (4Kw)
0x3FC000	reserved
0x3FE000	Boot ROM (8Kw)
0x3FFFC0	BROM Vectors (64w)
0x3FFFFF	Data | Program

Dual Mapped: L0, L1, L2, L3

CSM Protected: L0, L1, L2, L3, FLASH, ADC CAL, OTP

DMA Accessible: L4, L5, L6, L7, XINTF Zone 0, 6, 7

The F2833x can access memory both on and off the chip. The F2833x uses 32-bit data addresses and 22-bit program addresses. This allows for a total address reach of 4G words (1 word = 16 bits) in data space and 4M words in program space. Memory blocks on all F2833x designs are uniformly mapped to both program and data space.

The memory map above shows the different blocks of memory available to the program and data space.

The non-volatile internal memory consists of a group of FLASH-memory sections, a boot-ROM for up to 12 reset-startup options and a one-time-programmable (OTP) area. FLASH and OTP are usually used to store control code for the application and/or data that must be present at reset. To load information into FLASH and OTP, a dedicated download program is needed, which is also part of the Texas Instruments Code Composer Studio integrated design environment.

Volatile Memory is split into 10 areas called M0, M1 and L0 to L7 that can be used both as code memory and data memory.

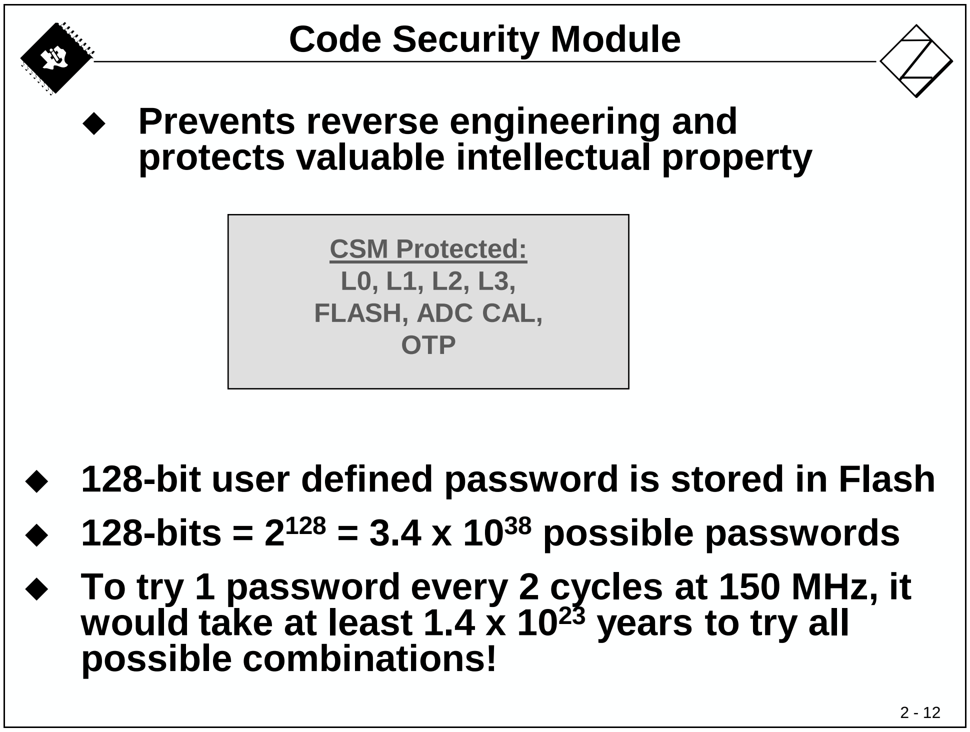
PF0, PF1 and PF2 are Peripheral Frames that cover control and status registers of all peripheral units ("Memory Mapped Registers").

Code Security Module

There is an internal security module available in all F2833x family members. It is based on a 128-bit password that is written by the ***software developer*** into the last 8 memory spaces of the internal FLASH (0x3F 7FF8 to 0x3F 7FFF). Once a pattern is written into this area, all further accesses to any of the memory areas covered by this Code Security Module (CSM) are denied, as long as the ***user*** does not write an identical pattern into password registers of frame PF0.

NOTE: If you write any pattern into the password area ***by accident***, there is no way to get access to this device anymore! Also, any attempt to re-flash a secured device will fail.

So please be careful and do not upset your laboratory technician!



Code Security Module

- **Prevents reverse engineering and protects valuable intellectual property**

CSM Protected:
L0, L1, L2, L3,
FLASH, ADC CAL,
OTP

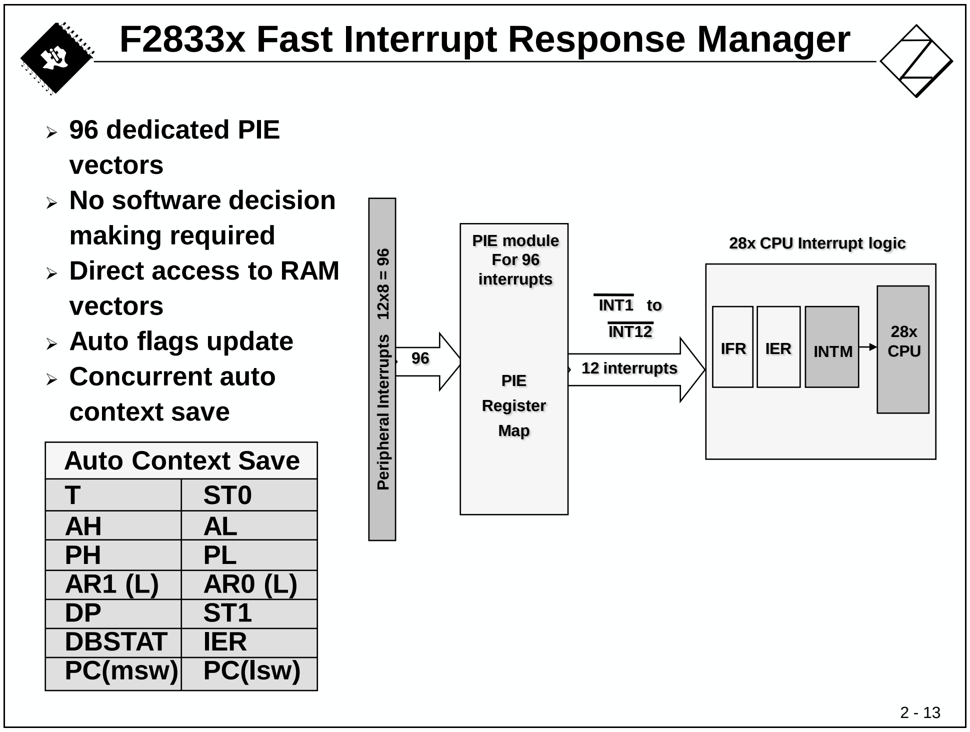
- **128-bit user defined password is stored in Flash**
- **128-bits = 2^{128} = 3.4 x 10^{38} possible passwords**
- **To try 1 password every 2 cycles at 150 MHz, it would take at least 1.4 x 10^{23} years to try all possible combinations!**

2 - 12

The purpose of a password secured device is to prevent reverse engineering of a control system. The code security module will deny any unauthorized access attempts and will protect your intellectual property (IP).

Interrupt Response

A key feature of a control system is its ability to respond to asynchronous external hardware events as quickly as possible. The F2833x combines such fast interrupt responses with an automatic "context" save of critical CPU registers, which allows the service of many asynchronous events with minimal latency. Here "context" means all the registers that need to be saved so that you can go away and carry out some other process, then come back to exactly where you left. F2833x devices implement a zero cycle penalty to save and restore the 14 registers during an interrupt. This feature helps to reduce the interrupt service routine overheads.



F2833x Fast Interrupt Response Manager

- 96 dedicated PIE vectors
- No software decision making required
- Direct access to RAM vectors
- Auto flags update
- Concurrent auto context save

Auto Context Save	
T	ST0
AH	AL
PH	PL
AR1 (L)	AR0 (L)
DP	ST1
DBSTAT	IER
PC(msw)	PC(lsw)

Peripheral Interrupts 12x8 = 96
96
PIE module For 96 interrupts
PIE Register Map
$\overline{\text{INT1}}$ to $\overline{\text{INT12}}$
12 interrupts
28x CPU Interrupt logic
IFR
IER
INTM
28x CPU

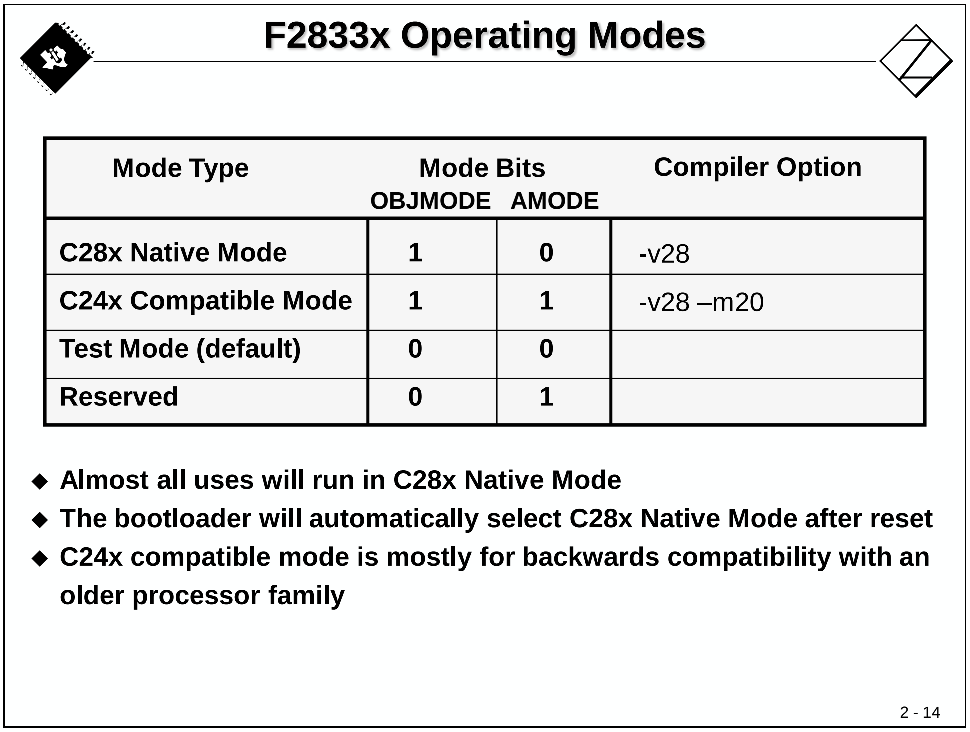
2 - 13

We will look in detail into the F2833x interrupt system in Module 6 of this tutorial. The Peripheral Interrupt Expansion (PIE) - Unit allows the user to specify individual interrupt service routines for up to 96 internal and external interrupt events. All possible 96 interrupt sources share 14 maskable interrupt lines (INT1 to INT14), 12 of them are controlled by the PIE - module.

The auto context save loads 14 important CPU registers, as shown in Slide 2-13 above, into a stack memory, which is pointed to by a stack pointer (SP) register. The stack is part of the data memory and must reside in the lower 64K words of data memory.

Operating Modes

The F2833x is a member of the TMS320C2000 family of Digital Signal Controllers (DSCs). This family consists both of 32-bit fixed-point and floating-point devices and also of 16-bit members. The Test Mode is used for fabrication test purposes only. The F2833x can be switched from its native mode into an operating mode, that is source code compatible with the 16-bit group C24x/C240x. Code, which has been previously written for a C24x device, can be reassembled to run on a F2833x device. This allows for migration of existing code onto the F2833x.



F2833x Operating Modes

Mode Type	Mode Bits		Compiler Option
	OBJMODE	AMODE	
C28x Native Mode	**1**	**0**	-v28
C24x Compatible Mode	**1**	**1**	-v28 –m20
Test Mode (default)	**0**	**0**	
Reserved	**0**	**1**	

- **Almost all uses will run in C28x Native Mode**
- **The bootloader will automatically select C28x Native Mode after reset**
- **C24x compatible mode is mostly for backwards compatibility with an older processor family**

2 - 14

In fact, the F2833x silicon is able to operate in three different modes:

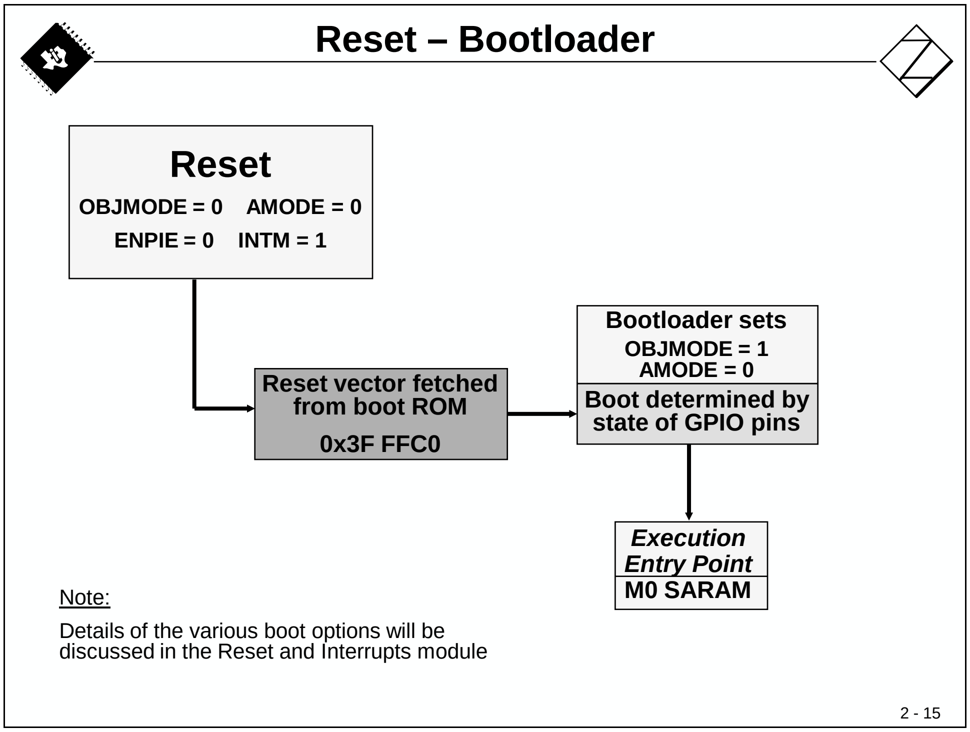
- C28x - Mode - takes advantage of all 32-bit features of the F2833x device
- C24x - Mode - source code compatibility to the 16-bit family members
- Test - Mode - intermediate operating mode, test purposes only.

After RESET, the device is set into test mode. To take advantage of the full computing power of an F2833x device, the control flag “OBJMODE” must be set to 1 to switch into F2833x native mode. If you start the execution of your code from the boot code entry point, the boot code will set that bit for you.

Reset Behaviour

After a valid RESET-signal is applied to the F2833x, the following sequence depends on some external pins on this DSC.

An active RESET signal will read the first address to be loaded into the Program Counter register (PC) from address 0x3F FFC0, which is in boot memory. The value inside this address is the address of the beginning of the boot code sequence. As a result, the F2833x jumps directly to the ***internal boot code*** memory. This code has been developed by TI to be able to distinguish between 12 different start options for the F2833x. The active option is derived from the status of 4 general-purpose input pins (GPIO) at this very moment. For our tutorial we use the volatile memory M0 as code memory and its first address as the execution entry point.



Reset – Bootloader

Reset
OBJMODE = 0 AMODE = 0
ENPIE = 0 INTM = 1

Reset vector fetched from boot ROM
0x3F FFC0

Bootloader sets
OBJMODE = 1
AMODE = 0

Boot determined by state of GPIO pins

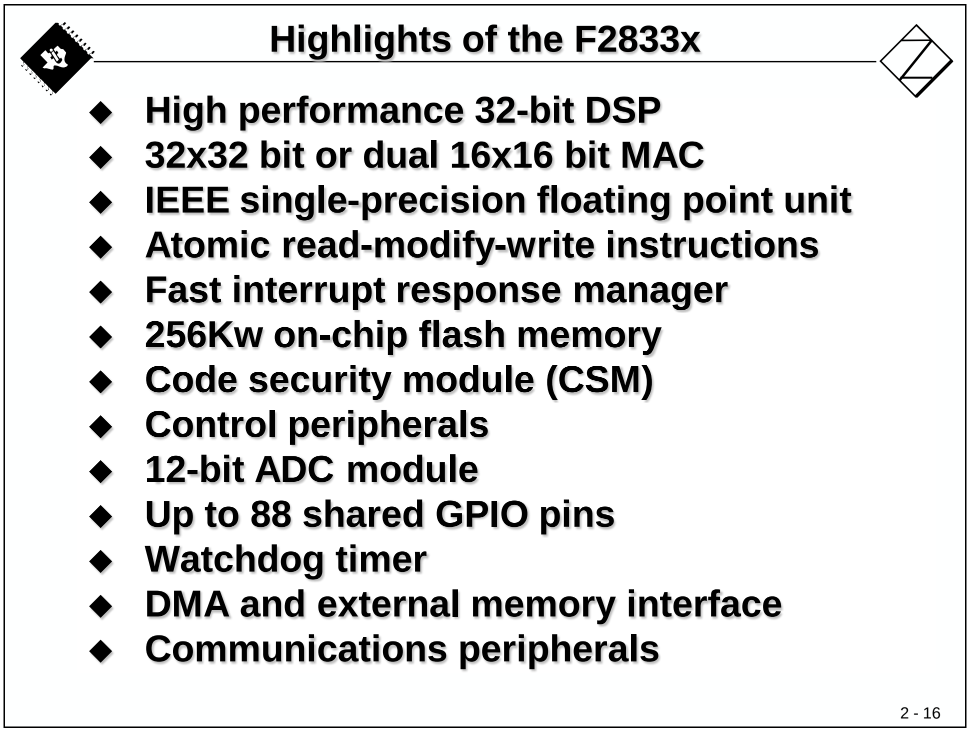
Execution Entry Point
M0 SARAM

Note:

Details of the various boot options will be discussed in the Reset and Interrupts module

2 - 15

Summary of TMS320F2833x Architecture



Highlights of the F2833x

- High performance 32-bit DSP
- 32x32 bit or dual 16x16 bit MAC
- IEEE single-precision floating point unit
- Atomic read-modify-write instructions
- Fast interrupt response manager
- 256Kw on-chip flash memory
- Code security module (CSM)
- Control peripherals
- 12-bit ADC module
- Up to 88 shared GPIO pins
- Watchdog timer
- DMA and external memory interface
- Communications peripherals

2 - 16

This page has been intentionally left blank.