Module Introduction

PURPOSE:

- This module, ColdFire MFC5249 Application-specific Peripherals, provides an overview of Freescale's ColdFire MFC5249 and the features and functions of its application-specific peripherals. More specifically, it will provide a baseline of knowledge related to the MFC5249 and its associated modules.

OBJECTIVES:

- Identify the features and functions of the Phase-Locked Loop (PLL).
- Identify the features and functions of the Analog-to-Digital Converter (ADC).
- Identify the features and functions of the Enhanced Multiply-Accumulator (EMAC) unit.
- Identify the features and functions of the IDE interface.
- Identify the features and functions of the FlashMedia interface.
- Identify the features and functions of the Audio Module interface.
- Identify the features and functions of the CD-ROM Block Decoder/Encoder.
- Identify the features and functions of the General Purpose Input/Outputs (GPIOs).
- Describe MCF5249 interrupt implementation.

CONTENT:

- 23 pages
- 4 questions

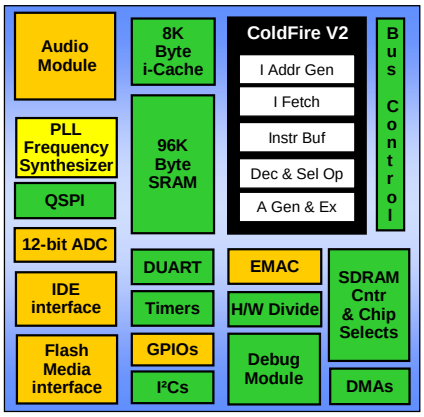
LEARNING TIME:

- 45 minutes

This module provides an overview of ColdFire MFC5249 and the features and functions of its application-specific peripherals. More specifically, it will provide a baseline of knowledge related to the MFC5249 and its associated modules.

In this module we will discuss the features and operation of the Phase-Locked Loop (PLL) and the Analog-to-Digital Converter (ADC). You will learn the attributes and functionality of the Enhanced Multiply-Accumulator (EMAC) unit. Next, you will examine the features and operations of several interfaces including the IDE interface, FlashMedia interface, and Audio Module interface. Then, you will review the CD-ROM Block Decoder/Encoders and General Purpose Input/Outputs (GPIOs) features and operations. Finally, you will learn the attributes and operations of MCF5249's interrupt implementation capabilities.

PLL Features



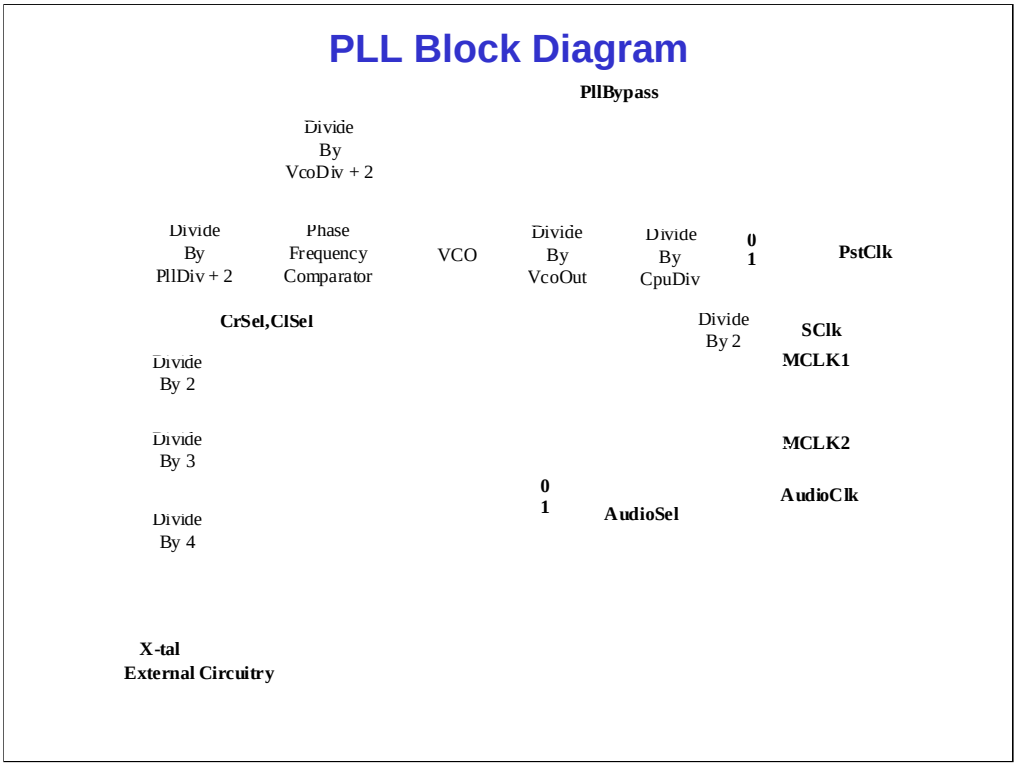
- **Clock Generator**
- **Bypass mode**
- **Configuration Utility**

Let’s start by looking at the Phase-Locked Loop (PLL).

The PLL and Clock Divider module lock to an external crystal clock frequency and generate clock signals for the processor, slave modules, and Audio Modules for the MCF5249.

The PLL also features a bypass mode. This mode allows the PLL to be configured by writing to a configuration register. By programming this register, the user may change the processor clock (PSTCLK) to a maximum of 140 MHz (160 MAPBGA) or a maximum of 120 MHz (144 QFP). The user can also bypass the PLL to allow the crystal clock to be fed directly to the processor clock.

A simple configuration utility for Windows that helps the user calculate the settings for the PLL configuration register for any desired processor frequency is available for download on the web site (www.freescale.com).



Let's take a closer look at the PLL. This figure shows the PLL module and the frequency relationships of various clock signals generated by the MCF5249 PLL. The PLL generates three clock signals.

The PLL generates the processor clock (PSTCLK).

The System clock (SCLK) is always one half of the PSTCLK. Its signals are fed to all of the slave modules, such as universal asynchronous receiver transmitters (UARTs) and timers.

The MCLK1 & MCLK2 clock signals are generated for the digital-to-analog converter (DAC).

The Audio Clock (AUDIOCLK) signals are generated for the Audio Modules.

MCF5249 PLL Configuration Wizard

- Win32 utility to calculate PLL frequencies and PLLCR settings
  - **Option 1: calculate PLLCR setting from target CPU frequency**
  - **Option 2: calculate CPU frequency from PLLCR settings**
  - **Option 3: calculate CPU frequency from individual PLLCR field settings**

The MCF5249 PLL Configuration Wizard is a Win32 utility provided to help with the configuration of the PLLCR for the MCF5249.

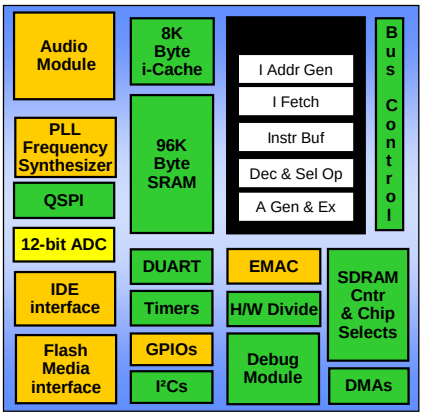
Three different options allow the user to quickly calculate the 32-bit PLL control register (PLLCR) register setting to achieve the desired frequencies produced by the PLL.

The first option calculates PLLCR settings from a target CPU frequency.

The second option calculates CPU frequency from PLLCR settings.

The third option calculates CPU frequency from individual PLLCR field settings.

Analog-to-Digital Converter



- **12-bit resolution**
- **Up to Four inputs**
  - ADC inputs are multiplexed
- **Internal digital portion**
- **Required external circuitry**

Next, let's take a look at the Analog-to-Digital Converter (ADC) and its functions and features.

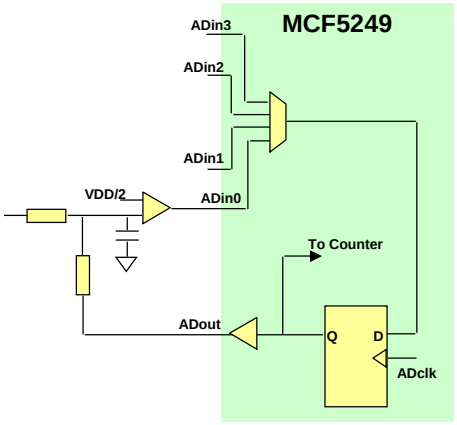
The ADC functionality on the MCF5249 is based on the Sigma-Delta concept using 12-bit resolution.

The ADC can have up to four inputs that are multiplexed.

The digital portion of the ADC is internal.

The analog voltage comparator must be provided from an external source. This will minimize Electromagnetic Interference (EMI) and help customize the analog circuitry.

12-bit ADC



- **Sigma-Delta concept**
- **External analog portion**
- **Supports four multiplexed inputs**
- **ADout provides reference voltage**
  - PDM format
  - External R-C integrator circuit required

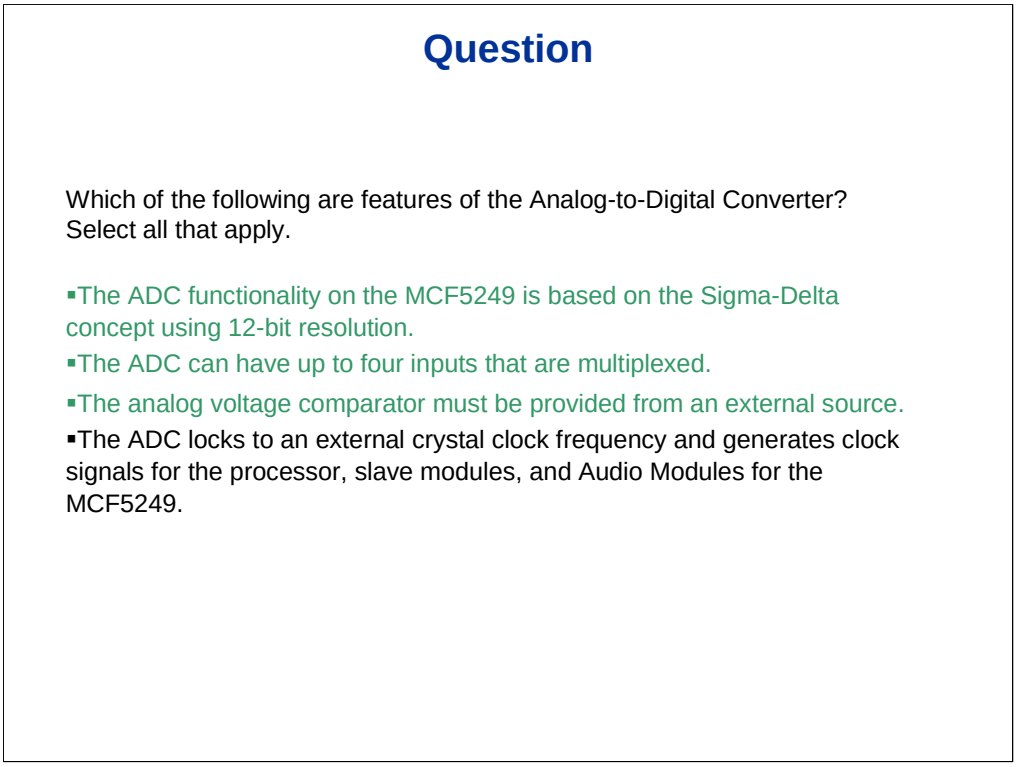
This figure shows an example of the external R-C integrator circuitry required for implementing a 12-bit ADC with the ADC module on the MCF5249.

The resistor-capacitor (R-C) integrator circuitry is based on the Sigma-Delta concept with 12-bit resolution.

The analog portion must be external (op amp or comparator).

The 12-bit ADC supports four multiplexed inputs.

AD out provides the reference voltage in PDM format. However, an external R-C integrator circuit is required.

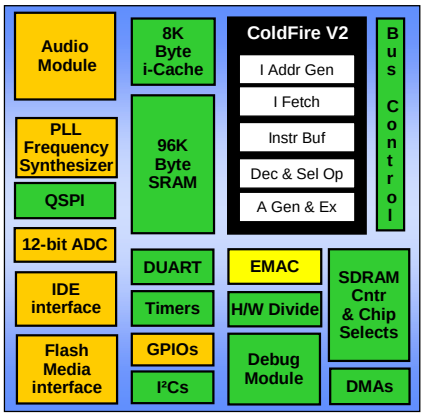


Consider this question about the Analog-to-Digital Converter. Select all that apply and then click Done.

Answer:

The ADC's functionality is based on the Sigma-Delta concept using 12-bit resolution. The ADC can have up to four inputs that are multiplexed and the analog voltage comparator must be provided from an external source.

EMAC



- **Four 48-bit accumulators**
- **Multiplication capabilities**
  - 40-Bit Products
    - 16x16
    - 32x32
- **Additional operands**
  - Signed Integers
  - Unsigned Integers
  - Signed, fixed-point fractions

The Enhanced Multiply-Accumulator (EMAC) unit is available on the MCF5249. EMAC is faster and more accurate for math-intensive software algorithms, especially with optimizing audio decoding/encoding (MP3).

See the following page for a direct comparison of the EMAC unit on the MCF5249 versus earlier ColdFire parts with the MAC unit.

The EMAC features four 48-bit accumulators.

The EMAC allows for 16x16 and 32x32 multiplies with a 40-bit product.

The EMAC unit provides functionality in three related areas. First, it provides signed and unsigned integer multiplies. It also provides miscellaneous register operations. Lastly, it provides multiply accumulate operations supporting signed and unsigned integer operands as well as signed, fixed-point, and fractional operands.

The EMAC unit also provides a four-stage execution pipeline, optimized for 32-bit operands, a fully pipelined 32 x32 multiply array and four 48-bit accumulators. Finally, it implements a 48-bit data path to allow the use of a 40-bit product plus the addition of eight extension bits.

EMAC vs. MAC

This is a direct comparison of the EMAC unit on the MCF5249 versus earlier ColdFire parts within the MAC unit.

EMAC

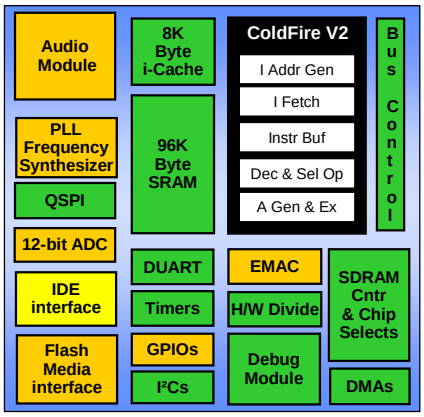
- **Four stage execution pipeline**
- **Optimized for 32-bit operands**
- **32x32 multiply array**
- **Four 48-bit accumulators**
- **40-bit products**

MAC

- **Three stage execution pipeline**
- **Optimized for 16-bit operands**
- **16x16 multiply array**
- **Single 32-bit accumulator**
- **32-bit products**

[Reference material for the previous page]

IDE interface

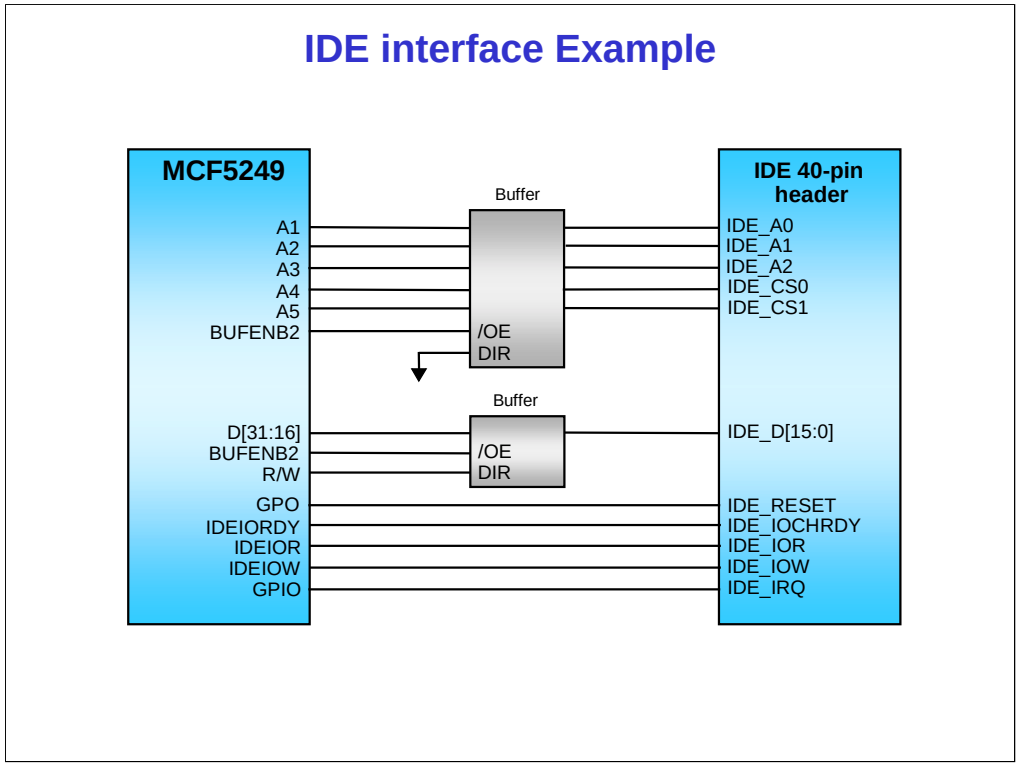


- **IDE connection**
- **MCF5249 CS2 used for IDE**
- **Buffer enable outputs supplied by MCF5249**

Next, let's look at the IDE interface and gather some details about its features and functions. The MCF5249 device system bus allows connection of an IDE hard disk drive with minimal external hardware. The IDE interface is useful in industrial and audio applications to interface to devices to store or retrieve large amounts of data in non-volatile memory such as a hard disk drive or CD-ROM.

The MCF5249 uses chip select 2 for control of the IDE interface. The IDE configuration registers are documented in the Chip Select chapter, section 10.5 of the user's manual.

The loading associated with the IDE bus will mean that buffers are required to reduce the loading on the MCF5249 bus. The MCF5249 also has two buffer enable outputs that help to eliminate the need for external logic to control address and data bus buffers. The enables are programmable to allow buffers to be connected singly or cascaded.



Let’s take a look at an IDE interface example. Notice that there is one set of buffers in the graphic set-up. The SDRAM or flash ROM is connected directly to the ColdFire bus. The IDE interface shares most signals with the ColdFire address and data bus. To prevent the flash ROM or SDRAM signals from going to or from the IDE interface, the address and data lines are buffered to the IDE interface. The BUFENB2 signal is used to control the /OE of the buffer.

IDE Setup Steps

1. **Program the Chip Select 2 registers.**
- **You need to program CSAR2, CSMR2, and CSCR2. Use the following settings for CSCR2:**
    - AA = 0 (/TA generated by IDEconfig2 register)
    - PS = 10 (16-bit port size)
    - BSTR, BSTW = 00 (disable bursting for read and write cycles)]

2. **Program the IDEconfig1 register.**
- **This step controls IDE bus timings and buffer enables.**

3. **Program the IDEconfig2 register.**
- **Use the following settings for IDEconfig2:**
    - TAenable2 = 1 (this allows the IDE control block to generate /TA for CS2 cycles)
    - IORDYenable2 = 1 (only if IORDY signal is used)]

Once the hardware interface is set up, there is a simple software sequence to configure the bus for the IDE interface.

First, write the proper Chip Select2 registers for the proper base address and bus size.

Next, write the IDEconfig1 register for bus timings and buffer enables.

Finally, write the IDEconfig2 register for the proper IDE control signals.

Question

Which of the following is a FALSE statement about Enhanced Multiply-Accumulator (EMAC) unit features and functions? Click on your choice and then select Done.

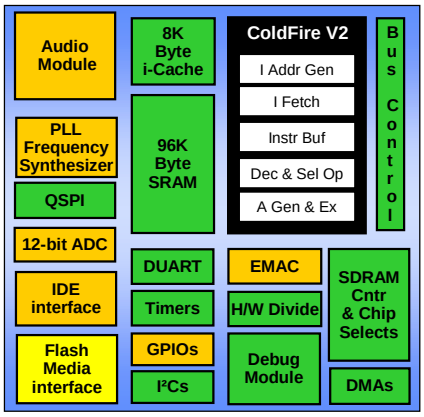
a) It contains a four-stage execution pipeline.
b) It can be optimized for 32-bit operands.
c) It only supports a 16x16 multiply array.
d) It contains four 48-bit accumulators.
e) It supports 40-bit products.

Consider this question about EMAC features and functions.

Answer:

The EMAC contains a four-stage execution pipeline, can be optimized for 32-bit operands, contains four 48-bit accumulators, and supports 40-bit products.

FlashMedia interface



- **Allows connection of FlashMedia cards with minimum external hardware**
- **MCF5249 CS3 used for FlashMedia**
- **Buffer enable outputs supplied by MCF5249**

Next, let's look at the FlashMedia interface and gather some details about its features and functions. The MCF5249 device system bus allows connection to FlashMedia cards such as the Sony Memory stick, with minimal external hardware.

The FlashMedia interface is configured similar to the IDE interface. The FlashMedia interface is controlled though chip select three (the IDE used chip select two). The two buffer enable outputs are also useful for reducing loading when the FlashMedia interface is used. Let's look at the details on the next page.

FlashMedia Setup

1. Program the Chip Select 3 registers (CSAR3, CSMR3, and CSCR3)
   - Use following settings for CSCR3
     - AA = 0 (/TA generated by IDEconfig2 register)
     - PS = 01 (8-bit port size)
     - BSTR, BSTW = 00 (disable bursting for read and write cycles)
2. Program the IDEconfig1 register
   - Controls IDE bus timings and buffer enables
3. Program the IDEconfig2 register
   - Use the following settings for IDEconfig2
     - TAenable3 = 1 (this allows the IDE control block to generate /TA for CS2 cycles)
     - IORDYenable3 = 1 (only if IORDY signal is used)
4. Program FlashMedia Control Registers

Once the hardware interface is set up, there is a simple software sequence to configure the bus for the FlashMedia interface.

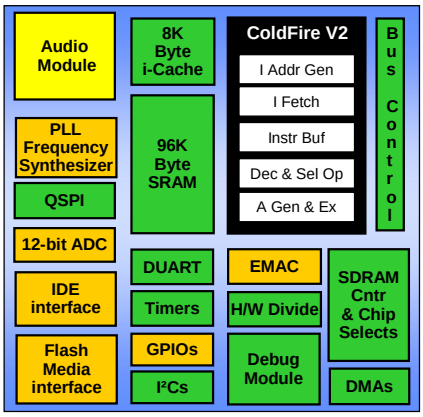
First, write the proper Chip Select registers for the proper base address and bus size of the FlashMedia interface.

Next, write the IDE Config1 register for bus timings and buffer enables.

Next, write the IDE Config2 register for the proper FlashMedia control signals.

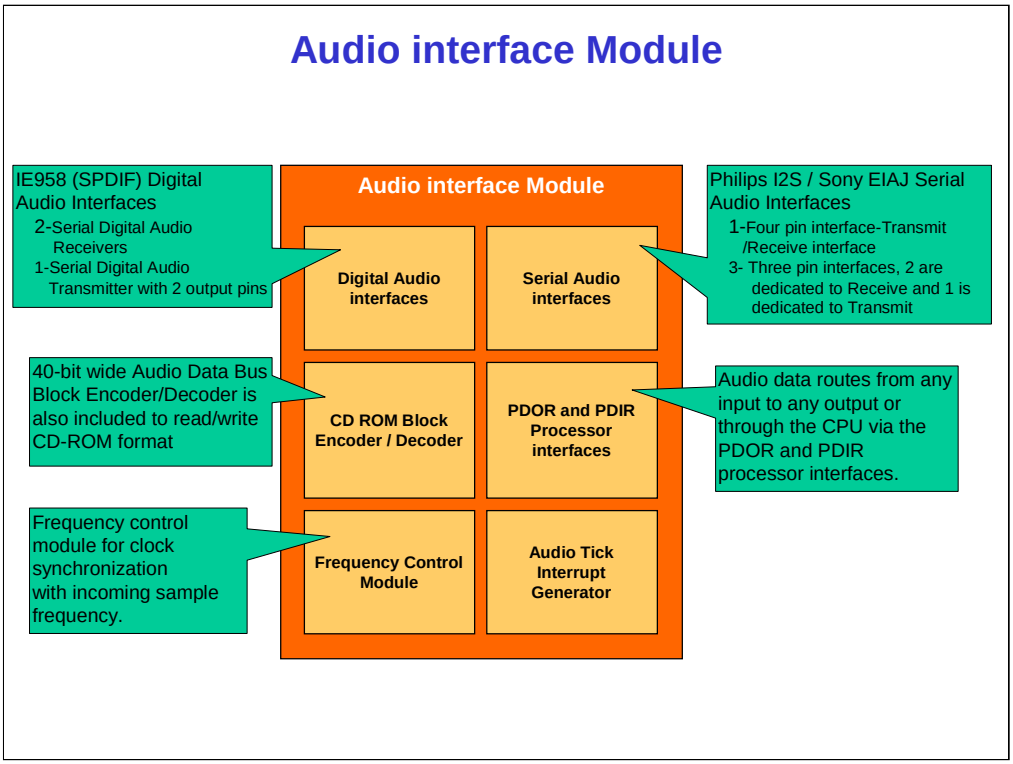
Finally, write the FlashMedia control register.

Audio Module



- Four serial audio interfaces supporting IIS and EIAJ formats
- Two IEC958 digital audio receivers with 4 muxed inputs
- One IEC958 transmitter with two outputs

Here we have the Audio Module interface, which is available on the MCF5249. This module allows the MCF5249 to receive and transmit digital audio over serial audio interfaces (IIS/EIAJ) and digital audio interfaces (IEC958). It contains four serial audio interfaces that support IIS and EIAJ formats.It also contains two IEC958 digital audio receivers with four multiplexed inputs. Finally, the last component of the Audio Module is one IEC958 transmitter with two outputs.



The Audio interface Module allows you to receive and transmit digital audio over serial audio and digital audio interfaces. The Audio Tick Interrupt is used to aid a busy system by allowing an interrupt to occur after a number of (programmable) sample pairs to help avoid under-run issues on the transmit or receive FIFOs.

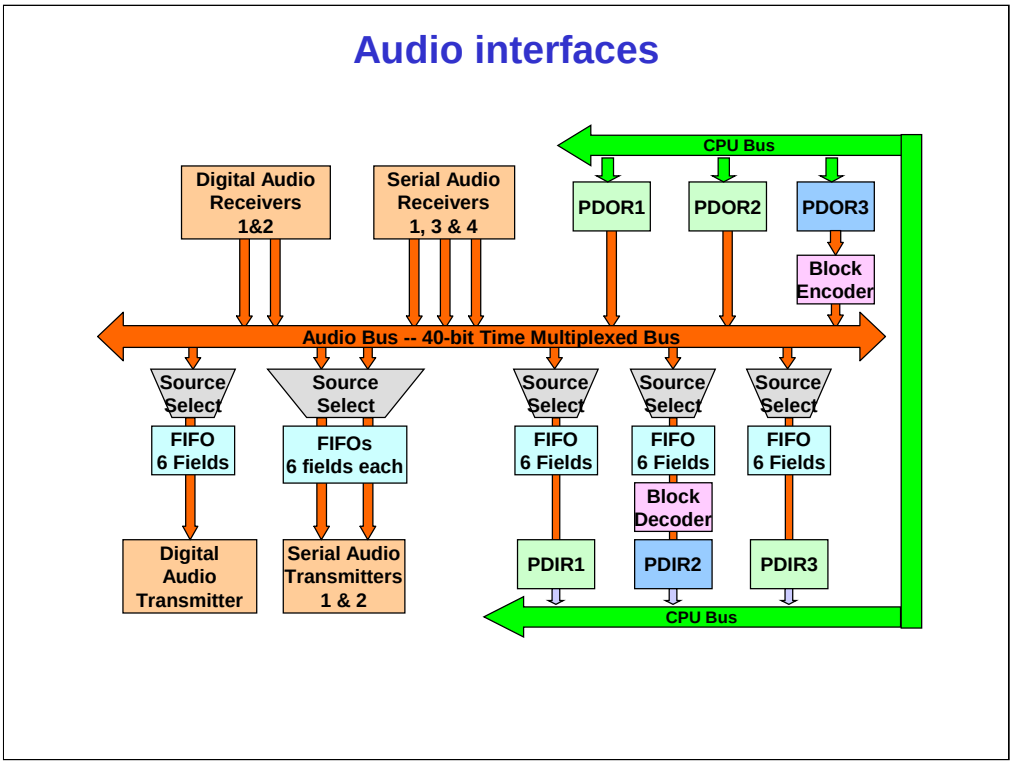
The Audio interface Module has two IEC958 digital audio input interfaces and one IEC958 digital audio output interface. There are four digital audio input pins and two digital audio output pins. An internal multiplexer selects one of the four inputs to the digital audio input interface. There is one digital audio output interface but it has two IEC958 outputs. One output carries the professional “c” channel and the other caries the consumer “c” channel. The remaining outputs carry identical data.

The module has four serial Philips I2S/Sony EIAJ audio interfaces. One interface is a 4-pin (one bit clock, one word clock, one data in, one data out). The other three interfaces are 3-pin (one bit clock, one word clock, one data in or out). The serial interfaces have no limit on minimum sampling frequency on bit clock input. This is one-third of the internal system clock.

The Audio interface Module also includes a 40-bit wide Audio Data bus CD ROM Block Encoder/Decoder to read and write CD-ROM format.

The module contains a frequency control module for clock synchronization with incoming sample frequency.

The PDOR and PDIR processor interfaces allow audio data routes from any input to any output or through the CPU.



Let’s take a closer look at the Audio interface Module’s internal structure. The Audio Module’s FIFOs (PDIR 1,2 and 3 and PDOR 1,2 and 3) interface directly to the internal ColdFire bus. Audio data sent to and from the FIFOs are routed to and from the specific audio interface through the 40-bit Audio bus controlled by the Audio interface Module’s configuration registers. Once the audio path is set up and configured, the user simply reads and writes the PDIR and PDOR registers respectively to receive and transmit audio data to and from the audio interfaces.

Question

Which of the following are Audio Module FIFOs?
Click on your choice and then select Done.

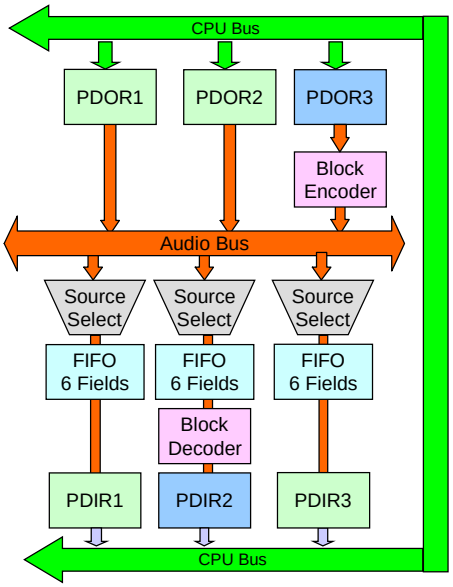
a) **PDOR1.**
b) **PDIR2**
c) **PDIR3**
d) **PDOR4**

Here’s a quick question for you.

Answer:

The Audio Module’s FIFOs (PDIR 1,2 and 3 and PDOR 1,2 and 3) interface directly to the internal ColdFire bus.

CD ROM Block Decoder/Encoder



- One 32-bit block control register is associated with the interface
- PDOR3 is equipped with a CD ROM block encoder
  - block by block basis
  - each block is generally 2352 bytes long
- PDIR2 is equipped with a CD ROM block decoder
  - Due to the addition of the decoder, PDIR2 is equipped with additional interrupts

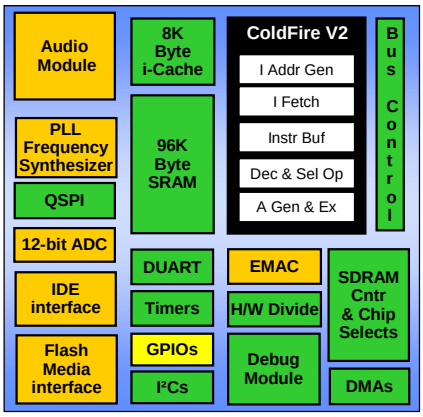
Let’s move on to the CD ROM Block Decoder/Encoder. This is used to interface to a CD ROM. This interface allows the MCF52429 to decode/encode CD ROM formats through the audio interface module.

One 32-bit block control register is associated with the interface.

PDOR3 is equipped with a CD ROM block encoder. This functions on a block by block basis, in which each block is generally 2352 bytes long.

PDIR2 is equipped with a CD ROM block decoder. Due to the addition of the decoder, PDIR2 is equipped with additional interrupts.

General Purpose I/Os

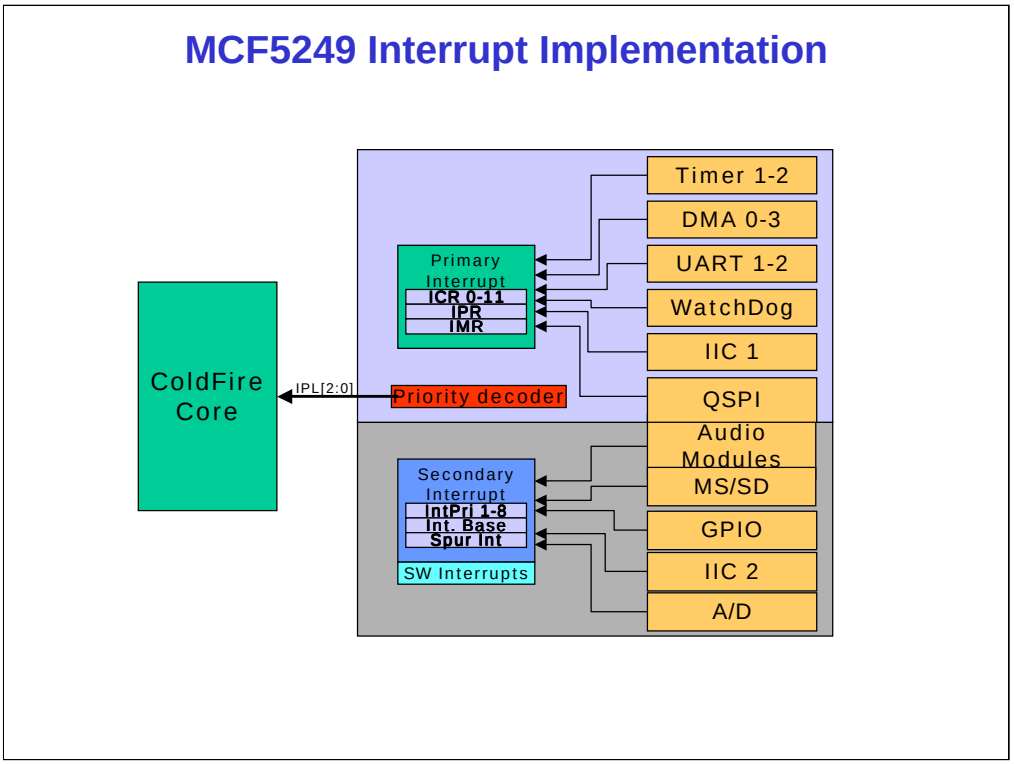


- Almost all peripheral pins can be configured for GPIO functionality
  - **Up to 47 GPIO pins**
  - **11 GPIs**
  - **13 GPOs**
- Edge sensitive interrupts on GPIO[7:0]

Let's take a look at General Purpose I/Os and their features and functions.

The majority of the MCF5249's peripheral pins can be configured for GPIO functions. Two groups of 32-bit registers control these GPIO pins. The MCF5249 160-pin package MAPBGA offers, 11GPI only pins, 13 GPO only pins and 47 combination GPIO pins.

The first eight GPIO's have edge-sensitive external interrupt capability on GPIO [7:0].



Finally, let's consider the MCF5249's interrupt implementation. Unlike previous ColdFire standard products, the MCF5249 offers two interrupt controllers: the Primary interrupt controller and Secondary interrupt controller.

The Primary interrupt controller is very similar to previous ColdFire standard products, code ported from early V2 Standard products, it can be "re-used".

The Secondary interrupt controller was added to handle interrupts from new audio, GPIO, ADC, and additional modules on the MCF5249.

Question

True or false: The Secondary interrupt controller was added to handle interrupts from new audio, GPIO, ADC, and additional modules on the MCF5249. Click on your choice.

TRUE

FALSE

Consider this question about MCF5249 Interrupt Implementation. Click the correct answer.

Answer:

The Secondary interrupt controller was added to handle interrupts from new audio, GPIO, ADC, and additional modules on the MCF5249.

Module Summary

- **Phase-Locked Loop (PLL)**
- **Analog-to-Digital Converter (ADC)**
- **Enhanced Multiply-Accumulator (EMAC) unit**
- **IDE interface**
- **FlashMedia interface**
- **Audio Module interface**
- **CD ROM Block Decoder/Encoder**
- **General Purpose I/Os (GPIOs)**
- **MCF5249 interrupt implementation**

Let's review the ColdFire MFC5249 Application-specific Peripherals that we examined in this module. First we learned about the features and operation of the Phase-Locked Loop (PLL) and the Analog-to-Digital Converter (ADC). We then looked at the attributes and functionality of the Enhanced Multiply-Accumulator (EMAC) unit. Next, we examined the features and operations of several interfaces including the IDE interface, FlashMedia interface, and Audio Module interface. Next, we reviewed the CD ROM Block Decoder/Encoder's and General Purpose Input/Output (GPIO) features and operations. Finally, we learned about the attributes and operations of MCF5249's Interrupt Implementation capabilities.