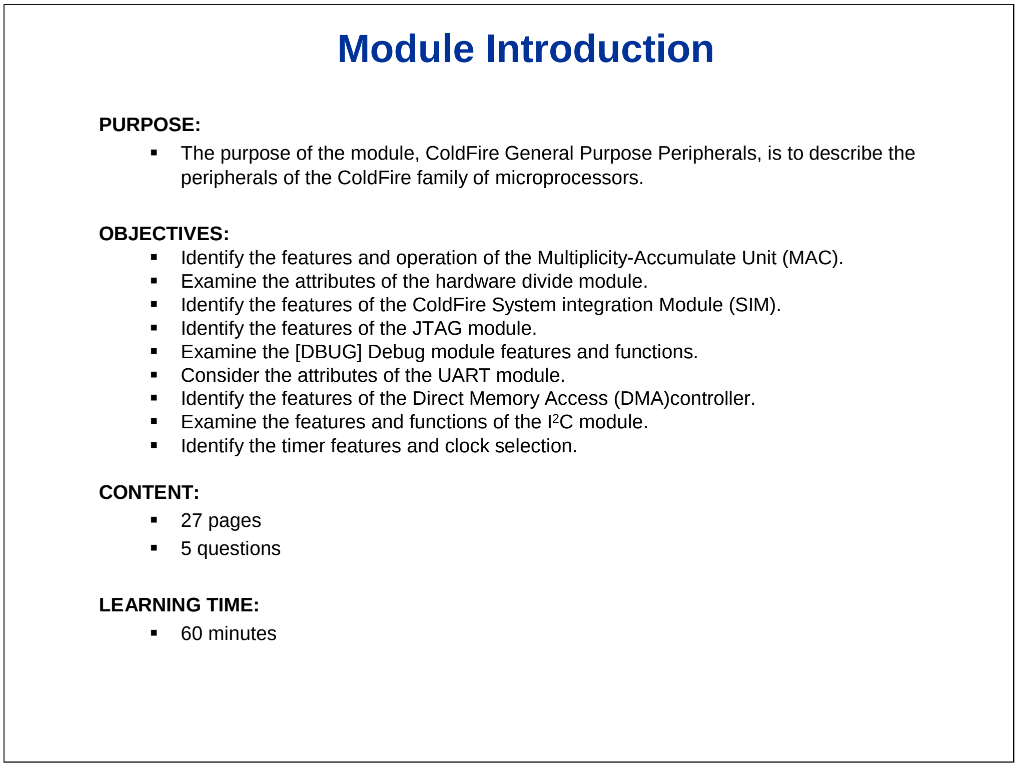


Module Introduction

PURPOSE:

- The purpose of the module, ColdFire General Purpose Peripherals, is to describe the peripherals of the ColdFire family of microprocessors.

OBJECTIVES:

- Identify the features and operation of the Multiplicity-Accumulate Unit (MAC).
- Examine the attributes of the hardware divide module.
- Identify the features of the ColdFire System integration Module (SIM).
- Identify the features of the JTAG module.
- Examine the [DBUG] Debug module features and functions.
- Consider the attributes of the UART module.
- Identify the features of the Direct Memory Access (DMA)controller.
- Examine the features and functions of the I^2C module.
- Identify the timer features and clock selection.

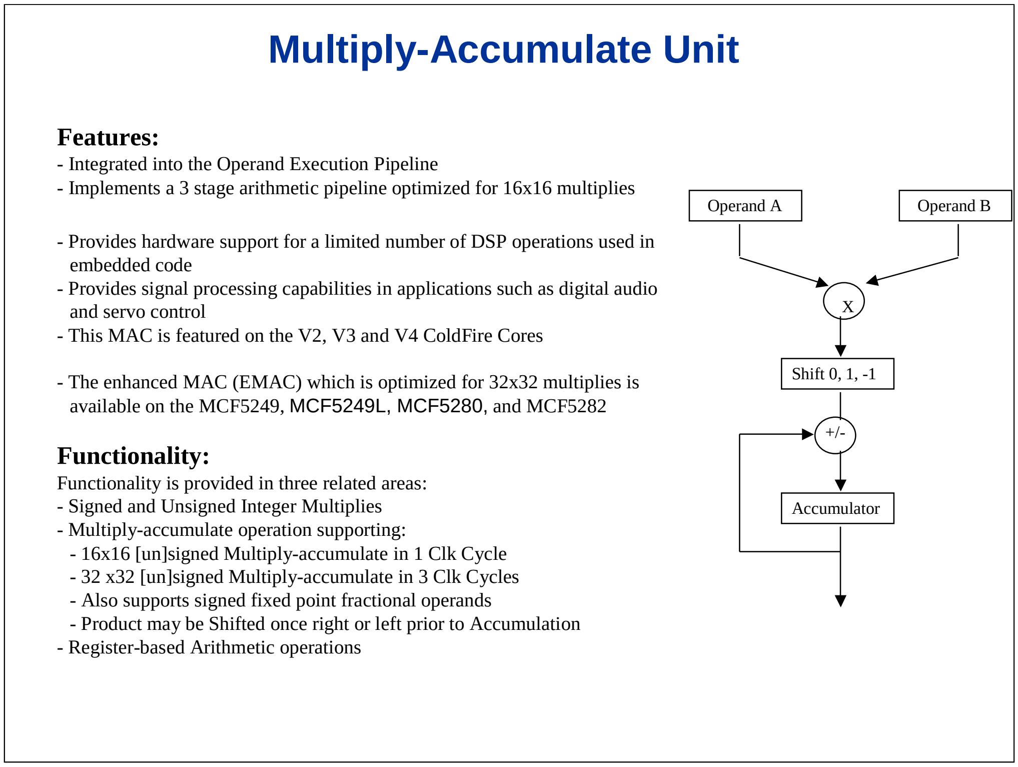
CONTENT:

- 27 pages
- 5 questions

LEARNING TIME:

- 60 minutes

The intent of the ColdFire General Purpose Peripherals module is to describe the peripherals of the ColdFire family of microprocessors. In this module, we will discuss the features and operation of the of the MAC programming model, the hardware divide module, and system integration. You will then look at the attributes and functionality of JTAG, [DBUG] Debug, and UART. Next, you will examine the features and operations of several controllers including the Direct Memory Access (DMA) and DRAM/SDRAM controllers. Finally, you will learn the features of the I^2C module and timer features and clock selection.



Multiply-Accumulate Unit

Features:

- Integrated into the Operand Execution Pipeline
- Implements a 3 stage arithmetic pipeline optimized for 16x16 multiplies

- Provides hardware support for a limited number of DSP operations used in embedded code
- Provides signal processing capabilities in applications such as digital audio and servo control
- This MAC is featured on the V2, V3 and V4 ColdFire Cores

- The enhanced MAC (EMAC) which is optimized for 32x32 multiplies is available on the MCF5249, MCF5249L, MCF5280, and MCF5282

Functionality:

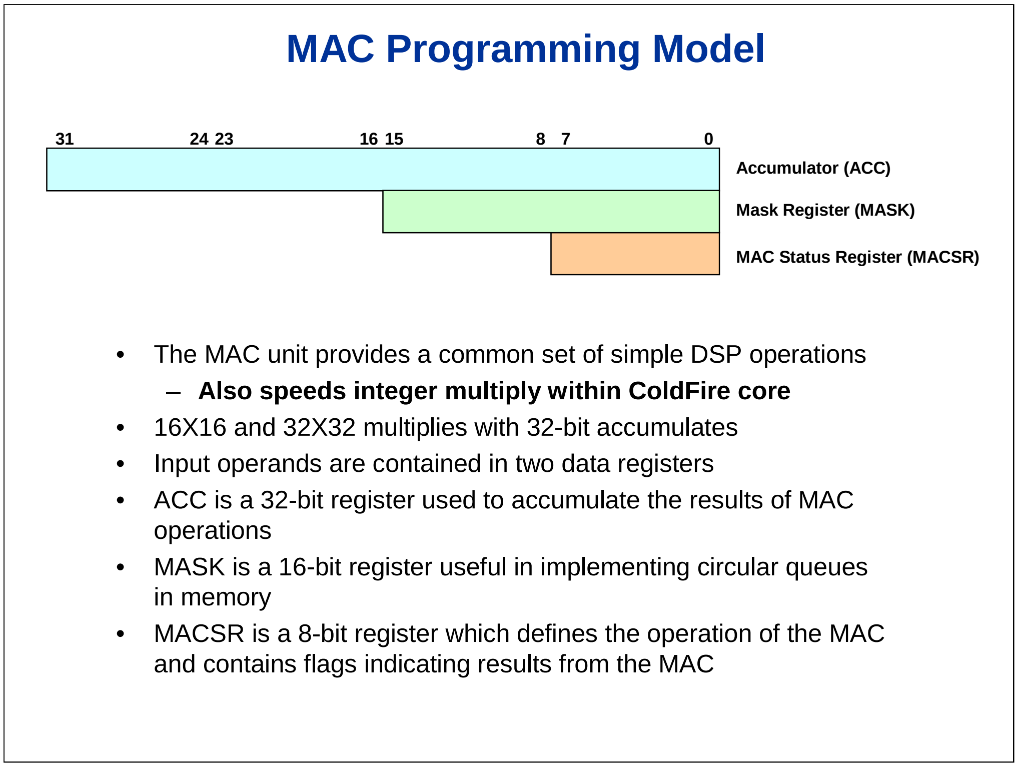
Functionality is provided in three related areas:

- Signed and Unsigned Integer Multiplies
- Multiply-accumulate operation supporting:
  - 16x16 [un]signed Multiply-accumulate in 1 Clk Cycle
  - 32 x32 [un]signed Multiply-accumulate in 3 Clk Cycles
  - Also supports signed fixed point fractional operands
  - Product may be Shifted once right or left prior to Accumulation
- Register-based Arithmetic operations

To begin, let’s look at the multiply-accumulate (MAC) unit. This unit provides hardware support for a limited set of digital signal processing (DSP) operations used in embedded code. The MAC supports the integer multiply instructions in the ColdFire microprocessor family.

The MAC unit is integrated into the Operand Execution Pipeline (OEP). This unit implements a three-stage arithmetic pipeline optimized for 16x16 multiplies. Both 16- and 32-bit operands are supported by this design in addition to a full set of extensions for signed and unsigned integers plus signed, fixed point fractional input operands. The MAC unit provides hardware support for a limited number of DSP operations used in embedded code. It also provides signal processing capabilities for ColdFire in a variety of applications including digital audio and servo control.

Note that this MAC is featured on the V2, V3, and V4 ColdFire cores. Also, the enhanced MAC (EMAC) which is optimized for 32x32 multiplies, is available on the MCF5249, MCF5249L, MCF5280, and MCF5282. The functionality of MAC is provided in three related areas.



MAC Programming Model

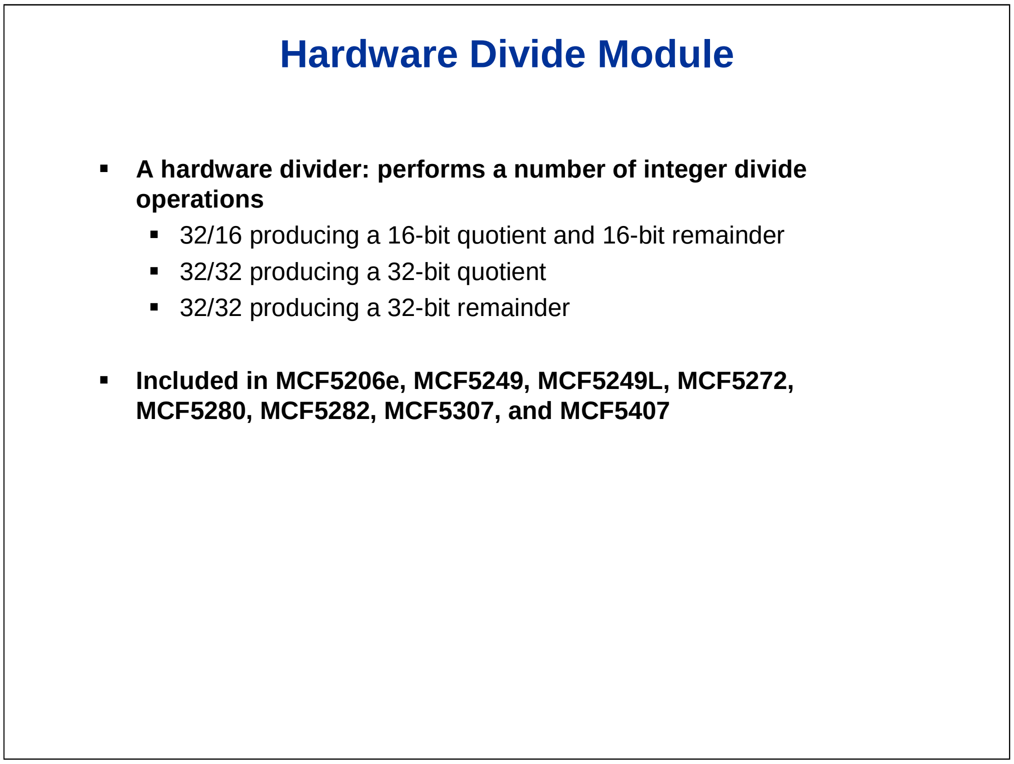
- The MAC unit provides a common set of simple DSP operations
  - **Also speeds integer multiply within ColdFire core**
- 16X16 and 32X32 multiplies with 32-bit accumulates
- Input operands are contained in two data registers
- ACC is a 32-bit register used to accumulate the results of MAC operations
- MASK is a 16-bit register useful in implementing circular queues in memory
- MACSR is a 8-bit register which defines the operation of the MAC and contains flags indicating results from the MAC

Here is the MAC programming model. It provides a common set of simple DSP operations, while speeding up integer multiplies within the Coldfire core. It supports 16X16 and 32x32 multiplies with 32-bit accumulates. The inputs are contained in two data registers.

The Accumulator (ACC) is a 32-bit register that is used to accumulate the results of MAC operations.

Mask register (MASK) is a 16-bit register that is useful in implementing circular queues in memory.

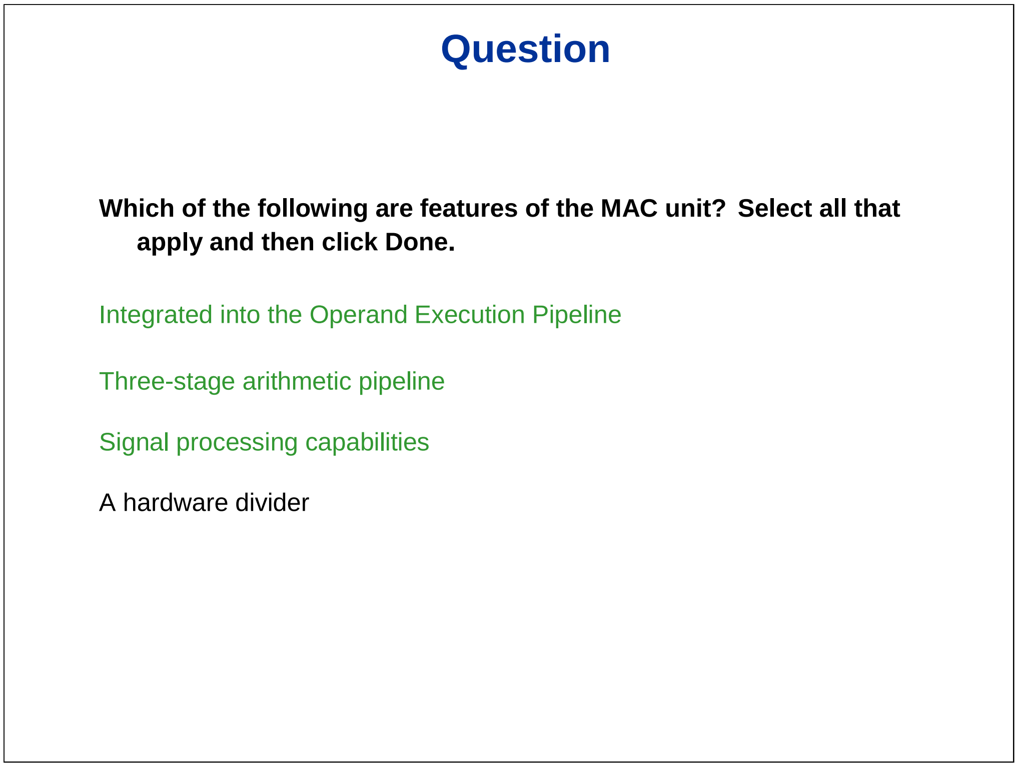
MAC Status Register (MACSR) is a 8-bit register that defines the operation of the MAC and contains flags indicating results from the MAC.



Hardware Divide Module

- **A hardware divider: performs a number of integer divide operations**
  - 32/16 producing a 16-bit quotient and 16-bit remainder
  - 32/32 producing a 32-bit quotient
  - 32/32 producing a 32-bit remainder
- **Included in MCF5206e, MCF5249, MCF5249L, MCF5272, MCF5280, MCF5282, MCF5307, and MCF5407**

Now, let’s look at the hardware divide module. Like the MAC unit, the hardware divide module is coupled to the core’s operand execution pipeline. It allows processors to support signed divides, unsigned divides, and remainder instructions. With this model, multiply and remainder instructions can take up to 38 clocks to execute. The actual execution time may be less depending on the addressing mode, operand size, and operand values.



Question

Which of the following are features of the MAC unit? Select all that apply and then click Done.

Integrated into the Operand Execution Pipeline

Three-stage arithmetic pipeline

Signal processing capabilities

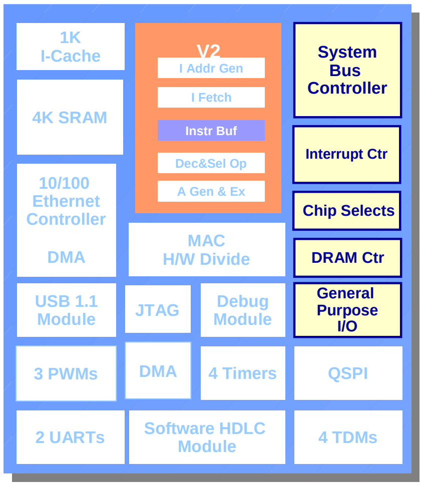
A hardware divider

Let’s review the features of the MAC unit with the following question.

Answer:

Operand execution pipeline, three-stage arithmetic pipeline, and signal capabilities are all features of the MAC unit.

System Integration Module



- Glueless bus interface with chip selects
- 8-, 16- & 32-bit support for DRAM, SRAM, ROM, FLASH & I/O devices
- 8 chip-select signals, 2 that are programmable with base address registers, 6 at an offset of one base register
- Programmable wait states & port sizes
- IEEE 1149.1 test (JTAG) compliant
- 16-bit general purpose I/O interface
- Programmable interrupt controller
  - Low interrupt latency
  - 4 external interrupt request inputs
  - Programmable autovector generator

Next is the System Integration Module (SIM). This module contains the System Bus Controller, Interrupt controller, Chip Select controller, DRAM controller and the General Purpose Input/ Output(GPIO) functions.

The Chip Select controller provides a glueless interface to most standard SRAMs, EPROMs, flash, and peripherals. Each of the eight chip select outputs has an address register, mask register, and burst capability. The mask register determines block size, and has the ability to address 8, 16, or 32-bit ports. The chip select outputs feature wait state generation and automatic acknowledge generation. They also feature address setup and address hold features.

Memory block sizes range from 64K to two gigabytes of address space. Chip select zero is active out of reset. Normally, this is where the reset vector and boot code reside. Initial parameters for chip select zero are loaded at reset, depending on the state of certain input pins. For example, on the MCF5206e, the state of IRQ1 and IRQ4 determine if chip select zero is an 8, 16, or 32-bit wide port.

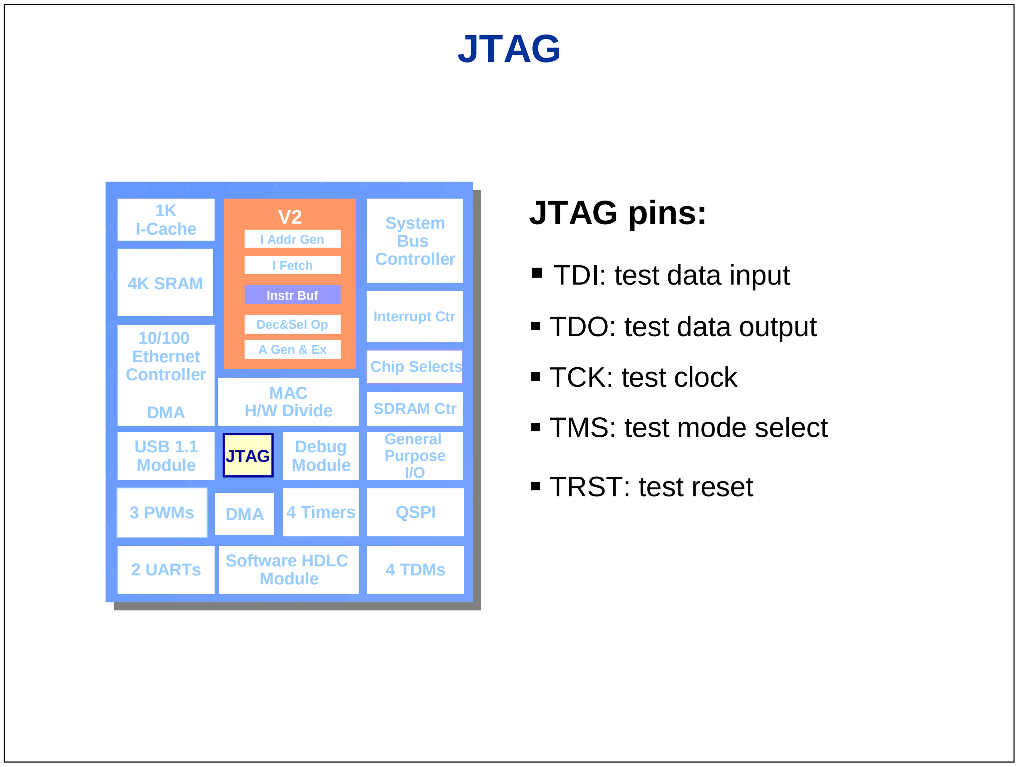
The SIM also features a parallel I/O port with a number of independent general purpose I/O pins. Depending upon the type of implementation, this port is either 8 or 16 bits wide. Each bit can be programmed individually as either an input pin or an output pin. Sometimes the pins have alternate functions. For example, they may be multiplexed with some debugging signals or upper address lines.

The next peripheral we'll look at is the interrupt controller, which resides in the SIM module. It has three external interrupt pins: IRQ1, IRQ4, and IRQ7. They can be programmed to individual interrupt requests at levels one, four, and seven. Level seven is non-maskable.

Internal interrupts can be programmed to any one of seven levels. No more than four interrupts can be assigned to a given level. Each interrupt at the same level must have a unique priority number from one to four assigned to it. You should not program interrupts to have the same level and priority; otherwise, unpredictable results will occur.

Seven distinct autovectors can be used, corresponding to the seven levels of interrupts. If autovector is disabled, then the external device or internal module must return the vector number during an interrupt acknowledge cycle.

Next, let's look at the test access port block diagram (JTAG).



JTAG

1K
I-Cache
4K SRAM
10/100
Ethernet
Controller
DMA
V2
I Addr Gen
I Fetch
Instr Buf
Dec&Sel Op
A Gen & Ex
MAC
H/W Divide
System
Bus
Controller
Interrupt Ctr
Chip Selects
SDRAM Ctr
USB 1.1
Module
JTAG
Debug
Module
General
Purpose
I/O
3 PWMs
DMA
4 Timers
QSPI
2 UARTs
Software HDLC
Module
4 TDMs

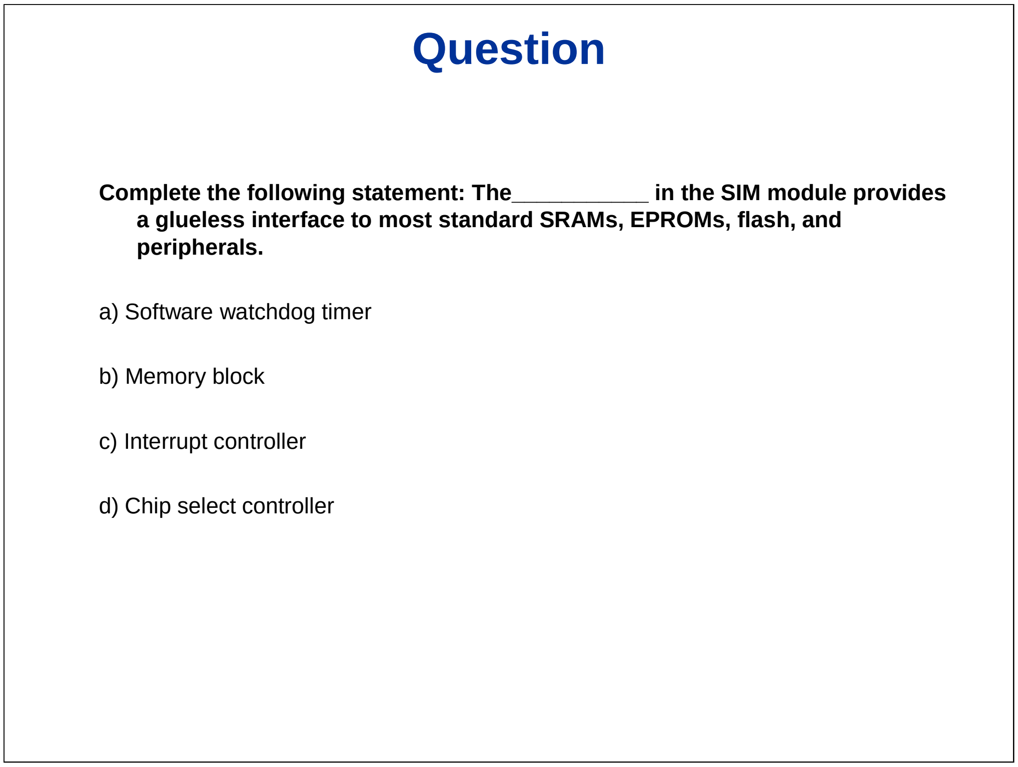
JTAG pins:

- TDI: test data input
- TDO: test data output
- TCK: test clock
- TMS: test mode select
- TRST: test reset

The JTAG interface is normally used to test board interconnects, which are a main cause of circuit board failure. It can also be used for some limited debug support. However, ColdFire devices that include background debug mode normally would not utilize this feature. JTAG eliminates the need and expense of a bed of nails board tester and supports circuit board test strategies that are based on the IEEE 1149.1 standard.

JTAG provides access to all of the data and chip control pins from the standard four-pin test access port and the active-low JTAG reset pin. One of the most commonly used features of JTAG is to perform boundary-scan operations to test circuit board electrical continuity. This function can detect open and short circuits on the circuit board for connections that interface to the ColdFire I/O. Another common use is to electrically isolate the ColdFire device from the circuit board. This is done using the HIGHZ instruction that will force all output and bidirectional pins into the high impedance state and protect the input-only pins from randomly toggling.

A useful register, accessible only through the JTAG port, is the ID code register. This is a JTAG compliant identification register that gives information about the ColdFire device. This register includes the device number, version number, and design center. It is useful to verify that the correct part and version number of the part is installed on the board. This information can be used to modify test routines or to load system software that pertains to the revision of the chip. JTAG allows exercising pins and capturing results without executing core logic on the CPU or other JTAG devices.



Question

Complete the following statement: The___________ in the SIM module provides a glueless interface to most standard SRAMs, EPROMs, flash, and peripherals.

a) Software watchdog timer

b) Memory block

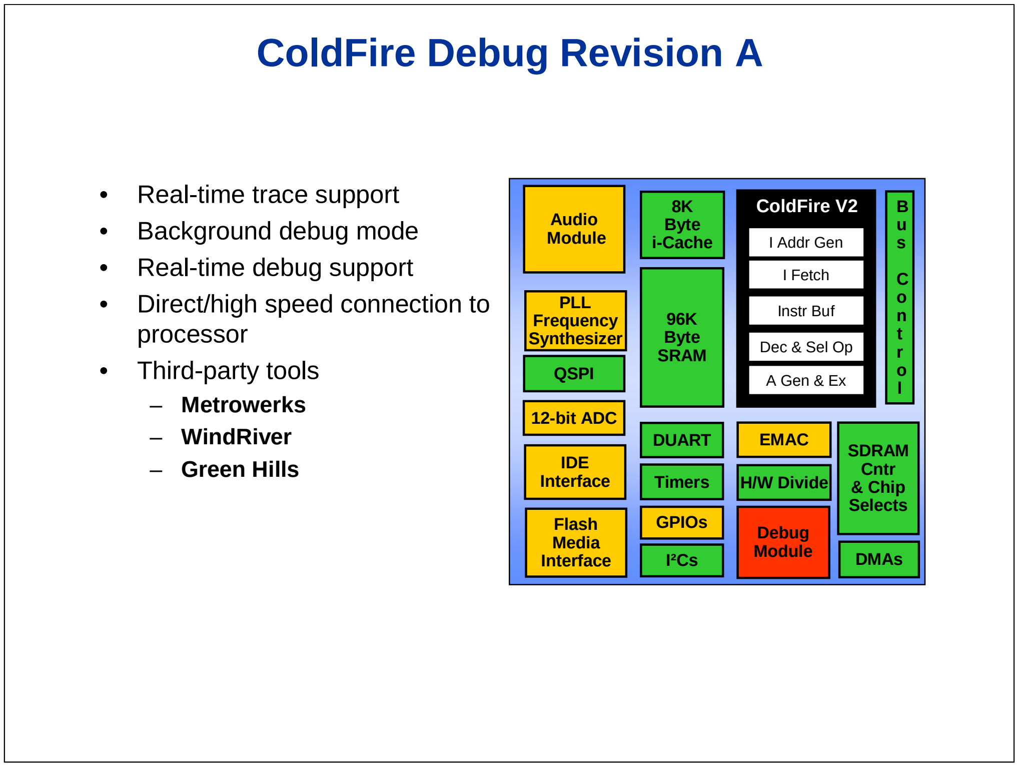
c) Interrupt controller

d) Chip select controller

Consider this question about the SIM module.

Answer:

It’s the chip select controller that provides a glueless interface to most standard SRAMs, EPROMs, flash, and peripherals.



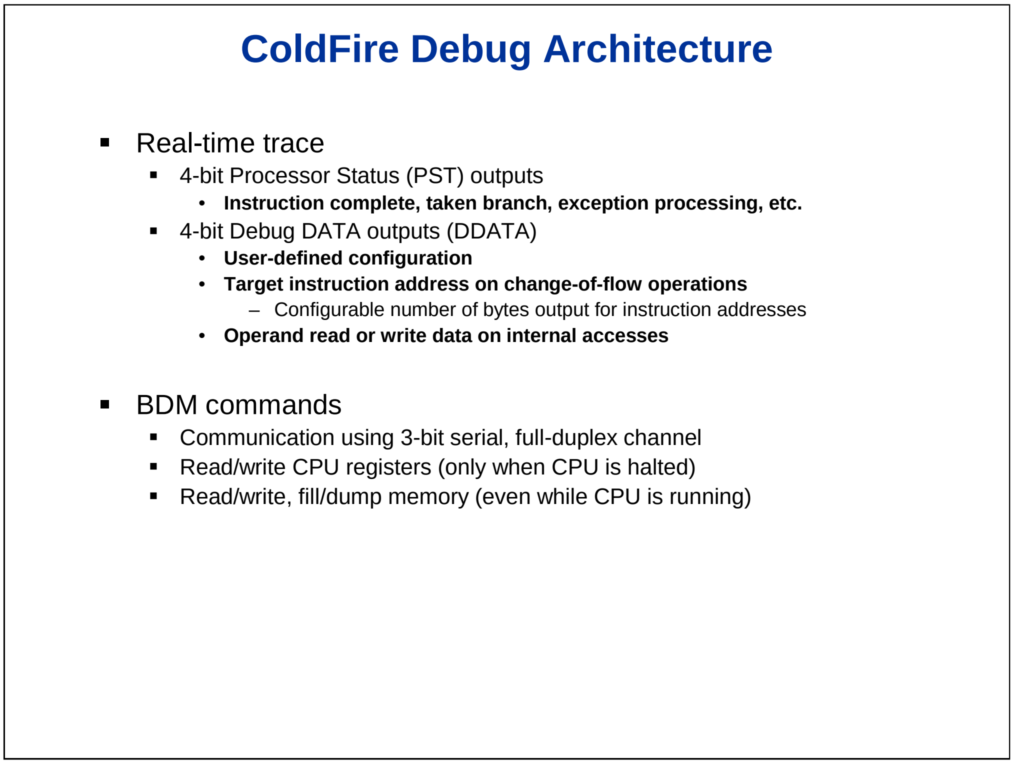
ColdFire Debug Revision A

- Real-time trace support
- Background debug mode
- Real-time debug support
- Direct/high speed connection to processor
- Third-party tools
  - **Metrowerks**
  - **WindRiver**
  - **Green Hills**

Audio Module
8K Byte i-Cache
ColdFire V2
I Addr Gen
I Fetch
Instr Buf
Dec & Sel Op
A Gen & Ex
Bus Control
PLL Frequency Synthesizer
96K Byte SRAM
QSPI
12-bit ADC
DUART
EMAC
SDRAM Cntr & Chip Selects
IDE Interface
Timers
H/W Divide
GPIOs
Flash Media Interface
Debug Module
I²Cs
DMAs

Now, let’s look at the ColdFire Debug, Revision A. Real-time trace support provides the ability to determine the dynamic execution path through an application. Background Debug Mode (BDM) provides low-level debugging in the ColdFire processor. In the BDM, the processor is halted and a variety of commands can be sent to the processor to access memory and registers.

Real-time debug supports allows for debugging without halting the processor. External third party tools communicate with the processor using the Debug module’s high speed three-wire serial interface. Debug tools for ColdFire are available from many vendors including Metrowerks, WindRiver, and Green Hills.

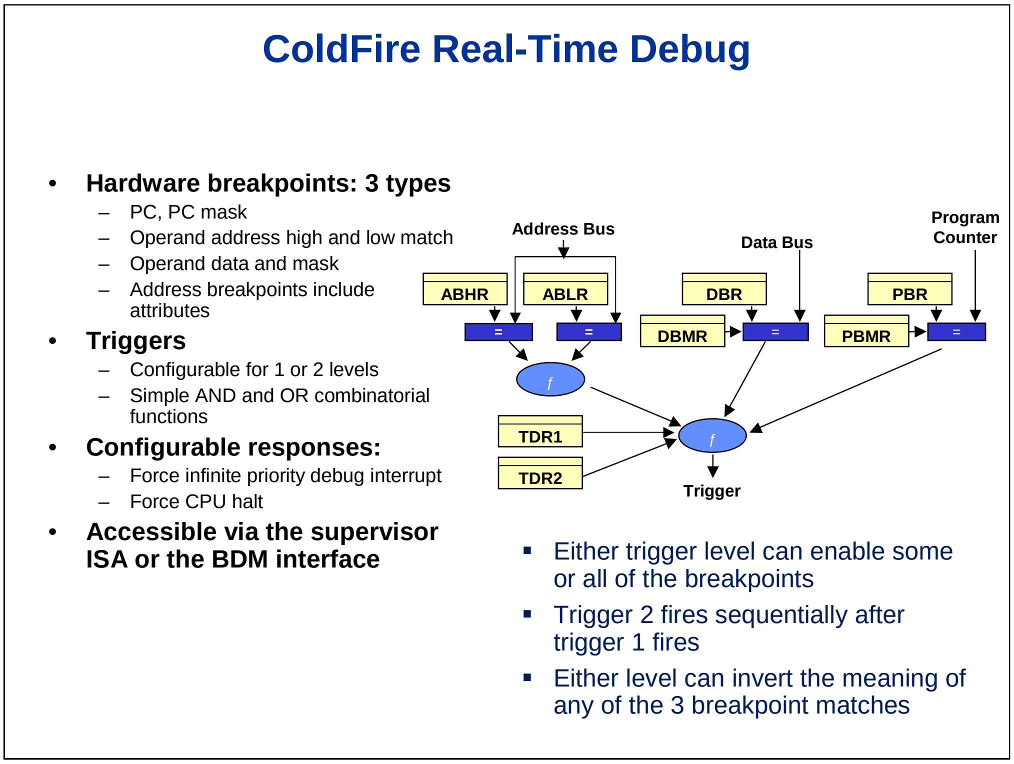


ColdFire Debug Architecture

- Real-time trace
  - 4-bit Processor Status (PST) outputs
    - **Instruction complete, taken branch, exception processing, etc.**
  - 4-bit Debug DATA outputs (DDATA)
    - **User-defined configuration**
    - **Target instruction address on change-of-flow operations**
      - Configurable number of bytes output for instruction addresses
    - **Operand read or write data on internal accesses**
- BDM commands
  - Communication using 3-bit serial, full-duplex channel
  - Read/write CPU registers (only when CPU is halted)
  - Read/write, fill/dump memory (even while CPU is running)

The ColdFire solution for real-time trace implements an 8-bit parallel output bus that reports processor execution status and data to an external emulator system. The information displayed on the DDATA lines is configurable. By default, the DDATA lines show the breakpoint status for the processor, but they can be programmed to display other information like branch target addresses and operands.

BDM commands are used to communicate between the processor and the third party debugger software. The 17-bit BDM command packets are transmitted via a 3-bit serial, full-duplex bus. BDM commands allow a debugger to read or write registers, control registers, and memory.

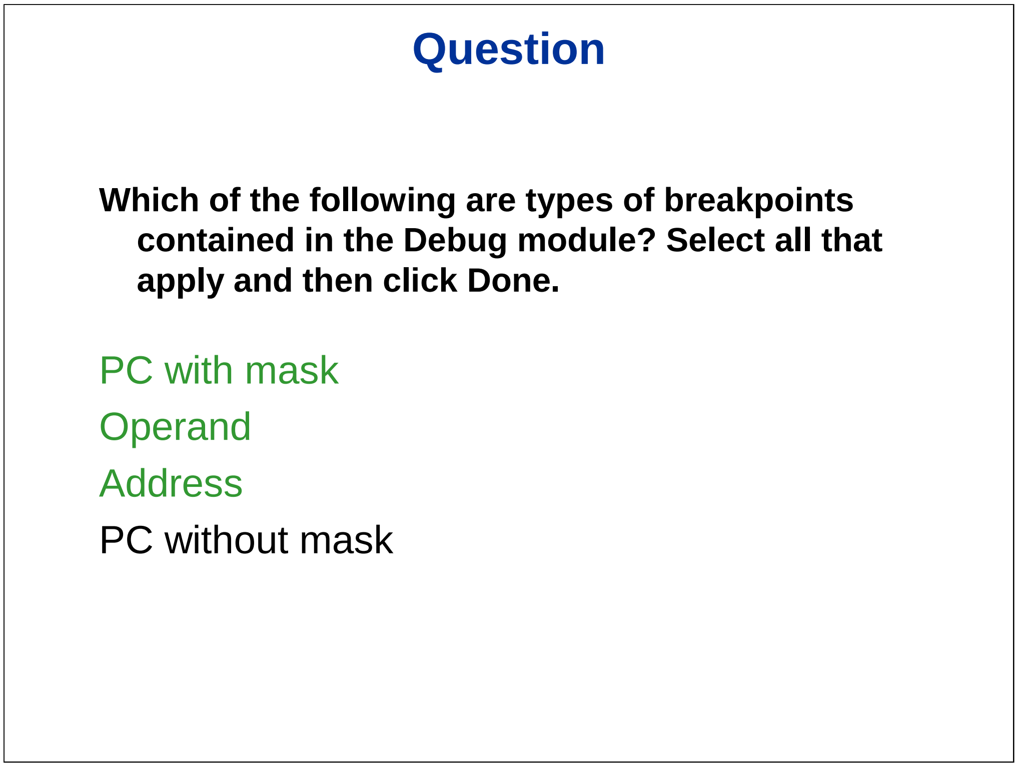


ColdFire Real-Time Debug

- **Hardware breakpoints: 3 types**
  - PC, PC mask
  - Operand address high and low match
  - Operand data and mask
  - Address breakpoints include attributes
- **Triggers**
  - Configurable for 1 or 2 levels
  - Simple AND and OR combinatorial functions
- **Configurable responses:**
  - Force infinite priority debug interrupt
  - Force CPU halt
- **Accessible via the supervisor ISA or the BDM interface**

- Either trigger level can enable some or all of the breakpoints
- Trigger 2 fires sequentially after trigger 1 fires
- Either level can invert the meaning of any of the 3 breakpoint matches

Next, let’s examine the ColdFire real-time debug. The debug module provides three types of breakpoints: PC with mask, operand, and address. These breakpoints can be configured into one- or two-level triggers. The exact trigger response is also programmable to either halt the CPU or process a debug exception. Either trigger level can enable some or all of the breakpoints. Trigger 2 fires sequentially after trigger 1 fires. Either level can invert the meaning of any of the 3 breakpoint matches.



Question

Which of the following are types of breakpoints contained in the Debug module? Select all that apply and then click Done.

PC with mask

Operand

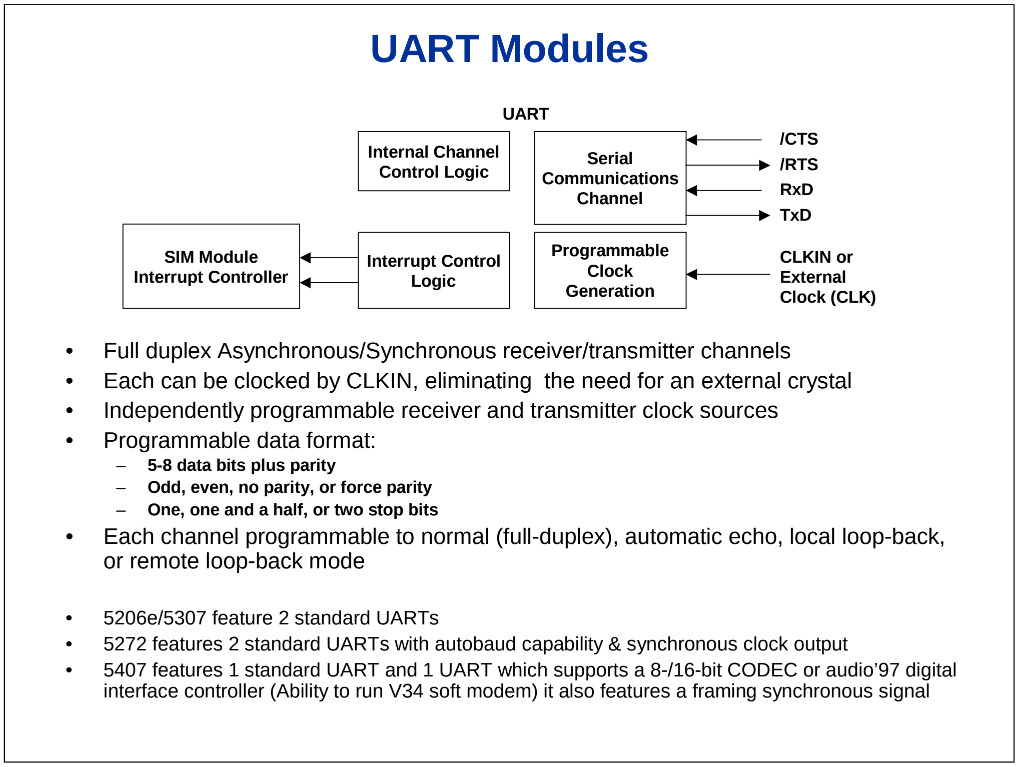
Address

PC without mask

Here is a question about Debug features.

Answer:

The debug module provides three types of breakpoints: PC with mask, operand, and address.



UART Modules

UART

Internal Channel Control Logic

Serial Communications Channel

/CTS

/RTS

RxD

TxD

SIM Module Interrupt Controller

Interrupt Control Logic

Programmable Clock Generation

CLKIN or External Clock (CLK)

- Full duplex Asynchronous/Synchronous receiver/transmitter channels
- Each can be clocked by CLKIN, eliminating the need for an external crystal
- Independently programmable receiver and transmitter clock sources
- Programmable data format:
  - **5-8 data bits plus parity**
  - **Odd, even, no parity, or force parity**
  - **One, one and a half, or two stop bits**
- Each channel programmable to normal (full-duplex), automatic echo, local loop-back, or remote loop-back mode

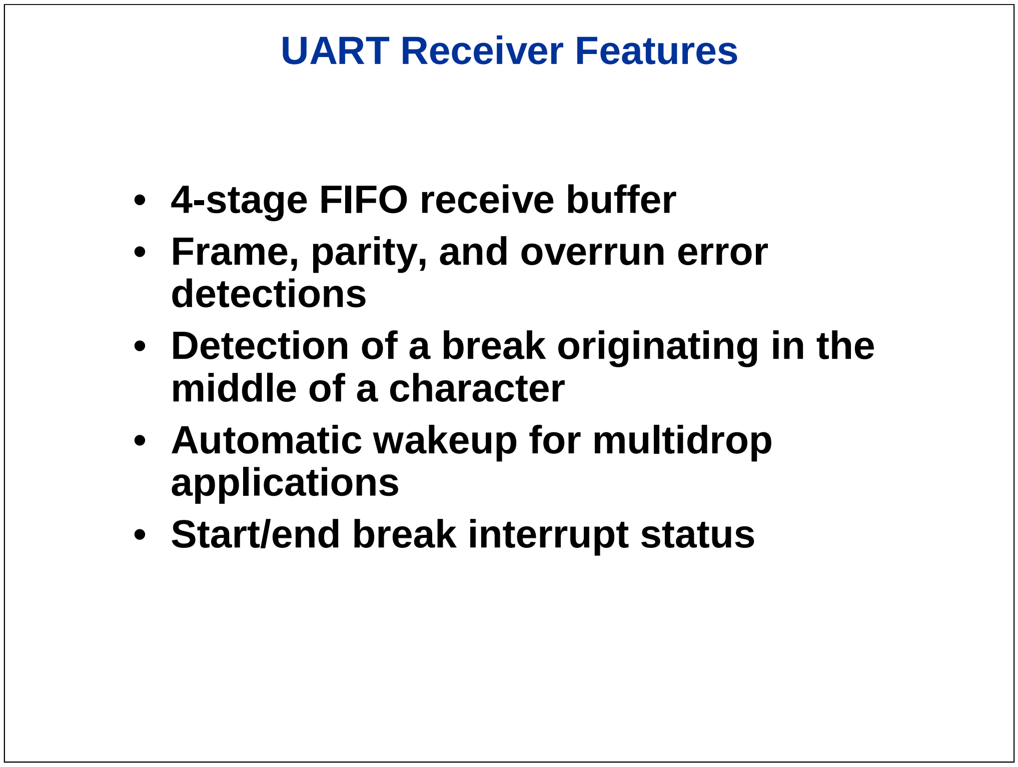
- 5206e/5307 feature 2 standard UARTs
- 5272 features 2 standard UARTs with autobaud capability & synchronous clock output
- 5407 features 1 standard UART and 1 UART which supports a 8-/16-bit CODEC or audio’97 digital interface controller (Ability to run V34 soft modem) it also features a framing synchronous signal

The next module we’ll examine is the universal asynchronous receiver/transmitter (UART). Each UART can be clocked by the system clock (CLKIN), which eliminates the need for an external UART crystal. UARTs can also be clocked by an external clock connected to the T in pin. There are four programmable channel modes. Normal mode will be discussed later. The three other modes are special cases that are used primarily for testing. These three modes are automatic echo, local loopback, and remote loopback.

Automatic echo mode takes the received data and retransmits it on a bit-by-bit basis. The local CPU-to-receiver communication continues normally but the CPU-to-transmitter link is disabled. Local loopback mode ties the receive and transmit pins internally together. This mode is useful for testing the operation of the local UART module channel by sending data to the transmitter and checking data assembled by the receiver. Both transmitter and CPU-to-receiver communications continue normally in this mode.

Remote loopback mode automatically retransmits received data just as automatic echo mode does except that both the receiver and transmitter are disabled from the CPU. This mode is useful for testing remote channel receiver and transmitter operation.

Let’s take a closer look at normal mode consisting of the receiver and transmitter.



UART Receiver Features

- 4-stage FIFO receive buffer
- Frame, parity, and overrun error detections
- Detection of a break originating in the middle of a character
- Automatic wakeup for multidrop applications
- Start/end break interrupt status

The UART receiver can be programmed to receive data formatted from 5 to 8 data bits plus a parity bit. Parity can be either odd, even or none. The receiver has a 4-character FIFO buffer. Interrupts can be generated when the first character is received or when the FIFO is full. Reading the FIFO will pop the stack, allowing space for

one more character to be received.

A status register contains information on error conditions, such as framing

error, parity error, and overrun error.

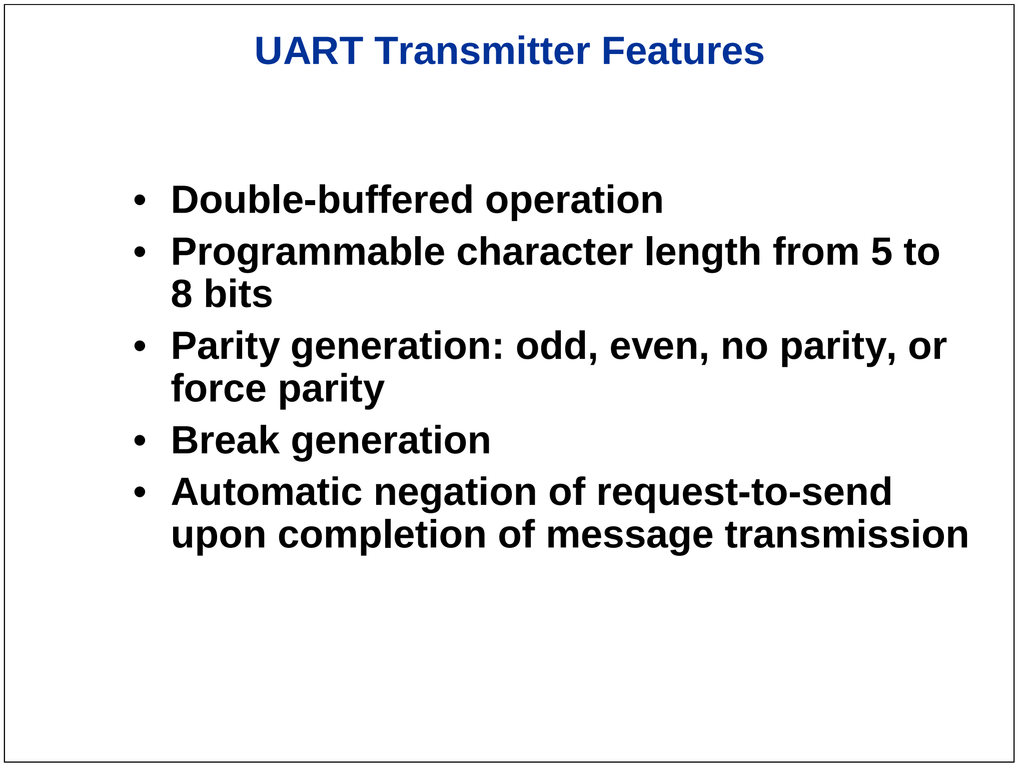
A bit for detecting a break character is also contained in this register. The

receiver operation can be either polled or interrupt driven.

You can program the receiver to operate in a wakeup mode for multidrop or multiprocessor applications. This mode of operation connects a master station to several slave stations. The slave stations’ UART receivers are disabled; however, they continuously monitor the data stream sent out by the master station.

When the master sends an address character, the slave receivers notify their respective CPUs by causing an interrupt. Each slave station CPU then compares the received address to its station address, and enables its receiver if it wants to receive the subsequent data stream.

Next, we’ll check out the UART transmitter.



UART Transmitter Features

- **Double-buffered operation**
- **Programmable character length from 5 to 8 bits**
- **Parity generation: odd, even, no parity, or force parity**
- **Break generation**
- **Automatic negation of request-to-send upon completion of message transmission**

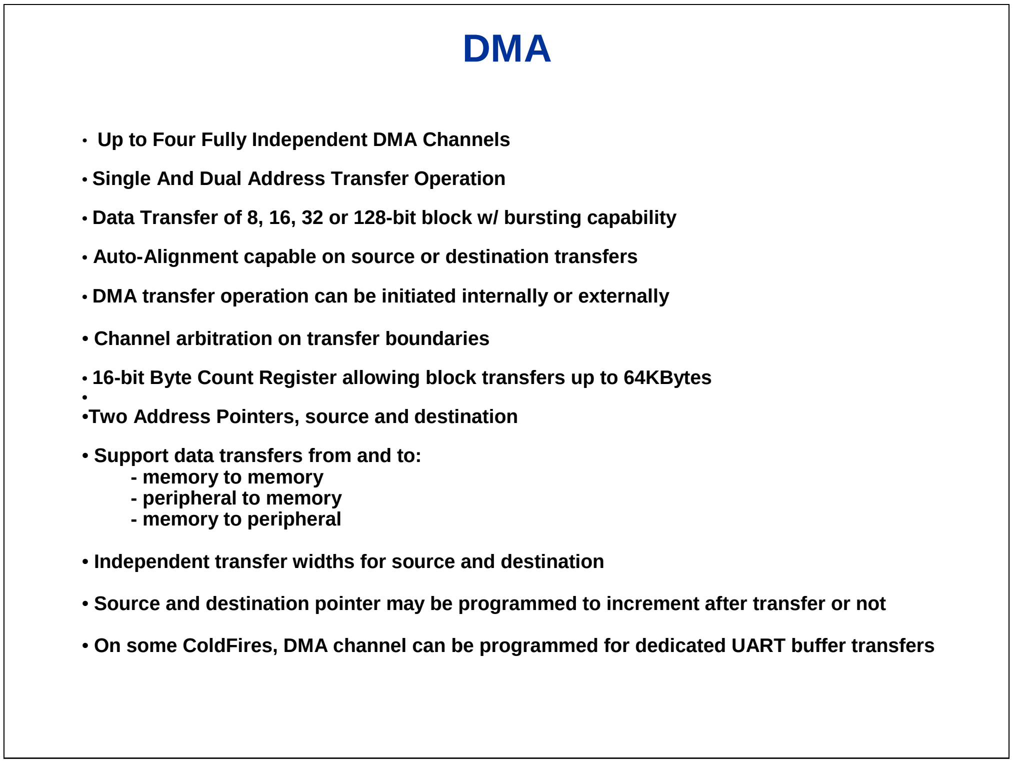
The UART transmitter is double buffered and can be programmed to transmit data formatted from 5 to 8 data bits plus parity. Parity can be either odd, even, none or forced. Forced parity simply means that the parity bit is forced either to a 1 or a 0, regardless of the actual parity. If selected, parity is automatically calculated by the

UART.

The UART has break character generation capability and can be programmed to send up to 2 stop bits. Full handshaking can be implemented with the request to send (RTS) and clear to send (CTS) pins.

You can program the transmitter to automatically negate the request to send output on completion of a message transmission. If clear to send is enabled, the CTS input must be asserted for the character to be transmitted. All standard baud rates up to 115K and higher can be achieved by using the internal system clock.

When using this clock, the formula to calculate the baud rate is: divide the system clock frequency by 32, and then divide the answer by the concatenation of the decimal value in the UBG1 and UBG2 registers. For example, to generate a baud rate of 19.2K with a 50 MHz system clock, you would program the UBG1 register to 0 and UBG2 register with decimal 81 or 51 hex.



DMA

- Up to Four Fully Independent DMA Channels
- Single And Dual Address Transfer Operation
- Data Transfer of 8, 16, 32 or 128-bit block w/ bursting capability
- Auto-Alignment capable on source or destination transfers
- DMA transfer operation can be initiated internally or externally
- Channel arbitration on transfer boundaries
- 16-bit Byte Count Register allowing block transfers up to 64KBytes
-
- Two Address Pointers, source and destination
- Support data transfers from and to:
  - memory to memory
  - peripheral to memory
  - memory to peripheral
- Independent transfer widths for source and destination
- Source and destination pointer may be programmed to increment after transfer or not
- On some ColdFires, DMA channel can be programmed for dedicated UART buffer transfers

The Direct Memory Access (DMA) controller module provides a quick and efficient process for moving blocks of data with minimal processor overhead. The DMA module provides up to four independent channels that allow byte, word, longword or 128 bit block with bursting capability for data transfers. These transfers can be single or dual address to off-chip devices or dual address to on-chip devices.

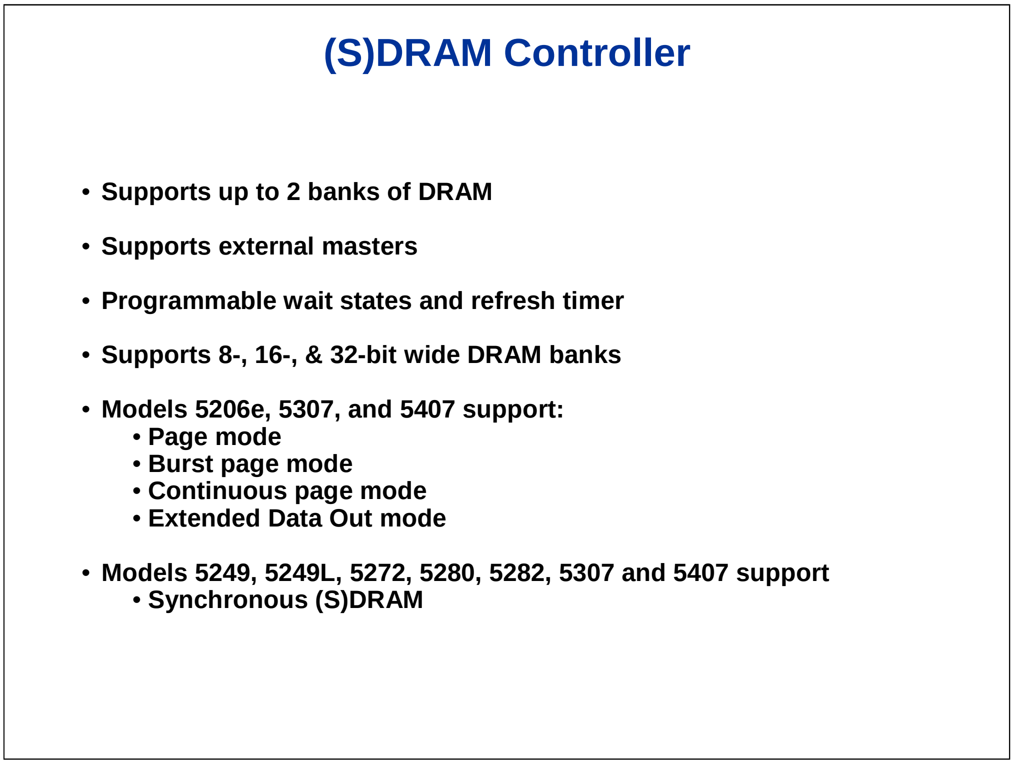
The DMA transfer operation can be initiated internally or externally. It can be initiated internally by setting the Start bit in the DMA control register, by pulling one of the two external DMA request pins low, or by the internal UARTs. Single address transfers take one bus cycle. Dual address transfers take two bus cycles. Channel arbitration takes place on transfer boundaries.

Depending on your memory system this can be as fast as two bus clocks for single address transfers and four bus clocks for dual address transfers. Glue logic is normally not required for dual address applications but is almost always required for single address applications. The block size is determined by the 16-bit value in the byte count register which will allow up to 64 thousand bytes per block to be transferred.

The DMA has two Address Pointers; source and destination. There are independent transfer widths for source and destination memory. The DMA supports auto alignment which means the accesses will be optimized based on the address value and the programmed size. The DMA supports data transfers to and from: memory to memory, peripheral to memory, and memory to peripheral.

Data packing and unpacking is also supported. This means that if the source and destination memory width is different, the bus accesses will accommodate the most efficient method to complete the transfer. For example, if you are reading from an 8-bit wide memory and writing to a 32-bit wide memory, the DMA will read the next four locations from the source memory and then make one write to the destination memory. This example assumes you are already memory aligned and you are not using burst memory. Source and destination pointer may be programmed to increment after transfer or not.

On some ColdFires, DMA channel can be programmed for dedicated UART buffer transfers



(S)DRAM Controller

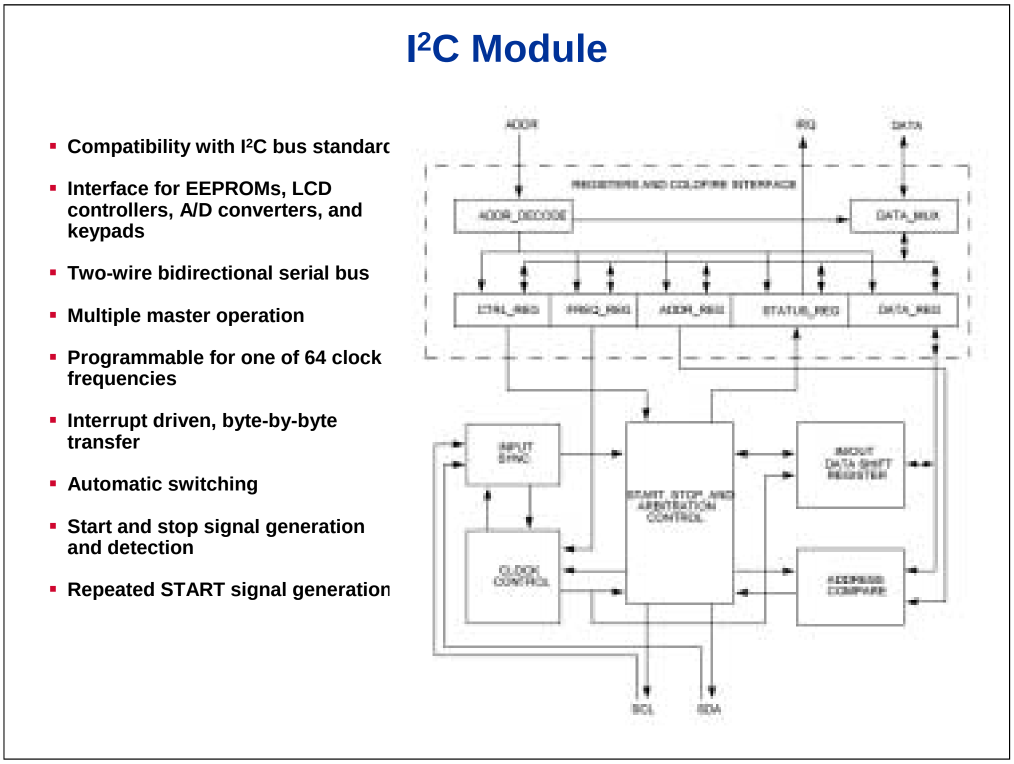
- Supports up to 2 banks of DRAM
- Supports external masters
- Programmable wait states and refresh timer
- Supports 8-, 16-, & 32-bit wide DRAM banks
- Models 5206e, 5307, and 5407 support:
  - Page mode
  - Burst page mode
  - Continuous page mode
  - Extended Data Out mode
- Models 5249, 5249L, 5272, 5280, 5282, 5307 and 5407 support
  - Synchronous (S)DRAM

The SDRAM controller provides control for /RAS, /CAS, and /DRAMW signals, as well as address multiplexing and bus cycle termination. The controller supports control signals and termination for both ADRAM or SDRAM. To reduce complexity, address line multiplexing is the same for both ADRAM and SDRAM operation. The DRAM can operate in burst or continuous mode.

The SDRAM controller module supports up to two banks of DRAM. These banks can be 8-, 16- or 32-bit wide DRAM. The module has programmable wait states and refresh timer.

Models 5206e, 5307, and 5407 support four modes: non-page, burst-page, continuous, and Extended Data Out (EDO). In non-page mode, the DRAM controller provides termination and runs a separate bus cycle for each data transfer. In burst-page mode, the row address remains registered while data is accessed from the different columns. Therefore only the first bus cycle in the page takes the full access time. Finally, continuous mode is a type of page mode that balances performance, complexity, and size. Page misses suffer no penalty. Finally, EDO mode allows the DRAM to continue driving data out of the device while /CAS is precharging.

The DRAM controller supports /CAS before /RAS refresh operations that are not synchronized to bus activity. In synchronous operation, SDRAM can be accessed on every clock after initial latency (typical 5,1,1,1 period). SDRAM's operate differently than asynch particularly in the use of data pipelines and commands to initiate special actions. Models 5249, 5249L, 5272, 5280, 5282, 5307, and 5407 support synchronous SDRAM.



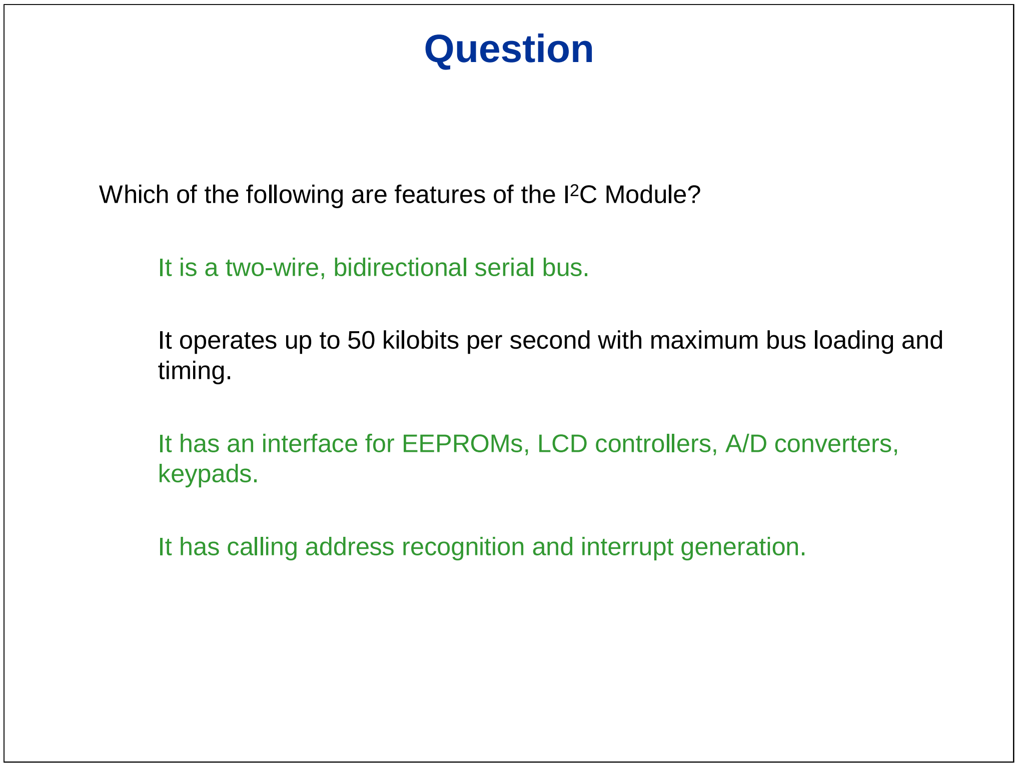
I²C Module

- Compatibility with I²C bus standarc
- Interface for EEPROMs, LCD controllers, A/D converters, and keypads
- Two-wire bidirectional serial bus
- Multiple master operation
- Programmable for one of 64 clock frequencies
- Interrupt driven, byte-by-byte transfer
- Automatic switching
- Start and stop signal generation and detection
- Repeated START signal generation

Now let’s examine the I²C Module. As shown in the diagram, it is a two-wire, bidirectional serial bus that provides a simple, efficient method of data exchange between devices. It is compatible with the widely used “I squared C” bus standard and is used as an interchip bus interface for devices such as EEPROMs, LCD controllers, A/D converters, and keypads.This two-wire multi-master bus minimizes the interconnection between devices and is suitable for applications requiring occasional communications over a short distance between many devices.

The I²C Module allows additional devices to be connected to the bus for expansion and system development. The I²C Module is a true multimaster bus including collision detection and arbitration that prevents data corruption if two or more masters attempt to control the bus simultaneously. The interface operates up to 100 kilobits per second with maximum bus loading and timing.

Some of the features include one of 64 serial clock frequencies that can be selected under software control, software selectable acknowledge bit, stop and start signal generation and detection, repeat start signal generation, interrupt on per byte transfer, and bus busy detection. In addition, the I²C Module has calling address recognition and interrupt generation and will automatically switch from master to slave on arbitration loss.



Question

Which of the following are features of the I^2C Module?

It is a two-wire, bidirectional serial bus.

It operates up to 50 kilobits per second with maximum bus loading and timing.

It has an interface for EEPROMs, LCD controllers, A/D converters, keypads.

It has calling address recognition and interrupt generation.

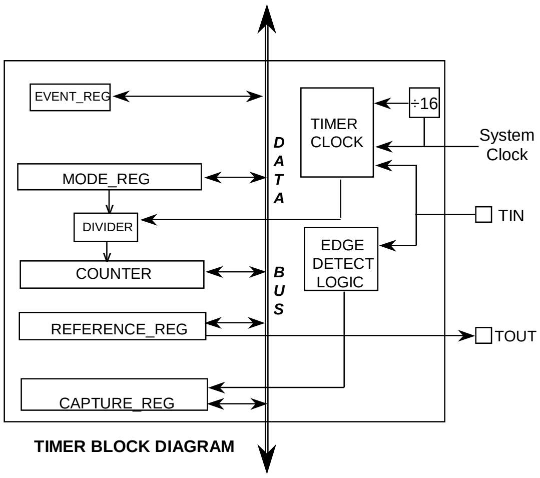
Consider this question about the I^2C module.

Answer:

I^2C’s features include two-wire serial bus, an interface for EEPROMs, LCD controllers, A/D converters, and keypads. It also has calling address recognition and interrupt generation.

Timer Features

- Two independent timer channels
- Output compare with programmable output signal
- Input capture with programmable edge
- Free run and restart modes

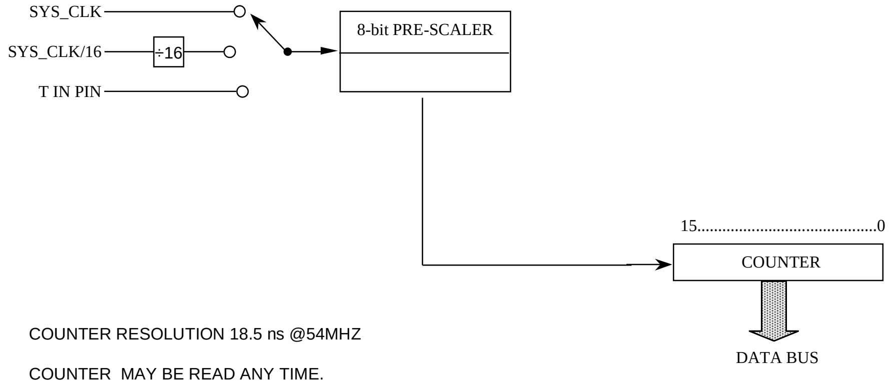


TIMER BLOCK DIAGRAM

The Coldfire family features independent 16-bit timers both programmable sources for the clock input, including an option for an external clock source. The timer can operate in a free running mode where the timer continues to run eventually overflowing back to zero and counting back up. Each timer has an input capture and output compare function with programmable modes for the input pin, T in, and the output pin, Tout.

In the next few pages, we will present these two functions and auto restart.

Timer Clock Selection



COUNTER RESOLUTION 18.5 ns @54MHZ

COUNTER MAY BE READ ANY TIME.

A WRITE TO THE COUNTER WILL RESET THE COUNTER

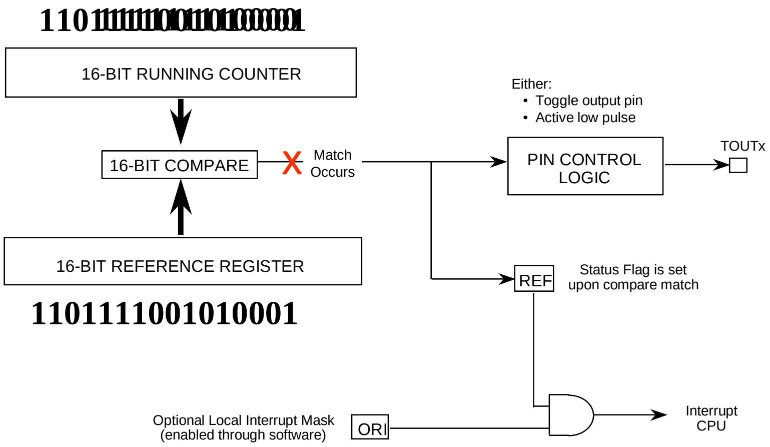
Let’s take a look at timer clock selection. The timer clock is designed to come from a variety of sources.

More specifically, the timer clock can be selected to come from the system clock, the system clock divided by sixteen, or the external timer input pin, T in.

This clock is then fed through an 8-bit prescaler. A prescaler value of zero will divide the clock by one where as a prescaler value of FF divides the clock by 256.

The 16-bit up counter register can be read at any time. However, a write to this register will reset the counter to zero. Another way to reset the counter is to clear the RST bit in the timer mode register. This not only will reset the counter to zero, it will also stop the timer from counting until this bit is set. The highest timer resolution is one system clock. For example, the Coldfire 5206e running at a system clock of 54 MHz can result in a timer resolution of 18.5 nanoseconds and a maximum period of five seconds.

Output Compare Function



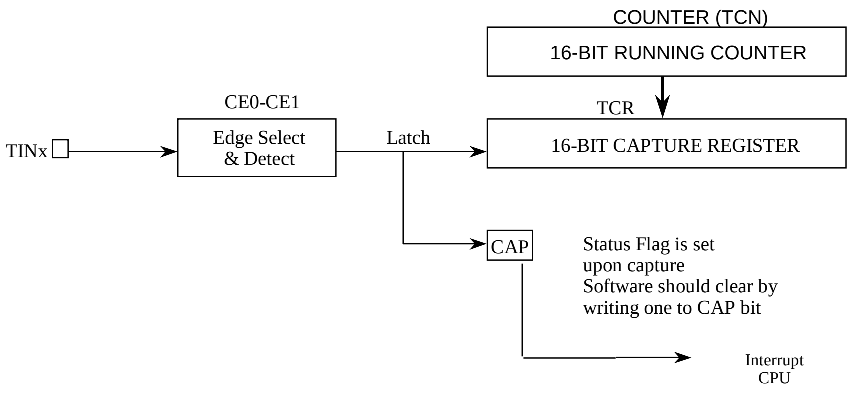
Let’s move on to the timer output compare function. This function provides a mechanism to output a signal at a specific time. It is very useful for generating pulse trains, one time pulse generation or single shot operation. It is also useful for generating periodic interrupts that are used to handle background tasks, such as scanning a keyboard, polling I/O or checking status flags.

The output compare logic contains a 16-bit reference register that compares each timer clock to the 16-bit running counter. If there is a match, the Tout pin will either toggle or produce an active low pulse for one system clock depending on the state of the OM bit in the timer mode register. The REF status bit is set and a timer interrupt occurs if enabled. At this point the 16-bit running counter will either continue counting up or reset to zero and start counting depending if you have the timer setup for free run or restart mode.

In order to use the timer in free run mode, the software will need to read the contents of the 16-bit running counter, add a value that represents a time that you want the event and/or interrupt to occur, and then write this value into the 16-bit reference register. When there is a match the event, Tout, will occur and, if enabled, an interrupt will be generated.

In some cases, the function of the output pin, Tout, is not used. For example, to generate a periodic interrupt you need to set the timer to restart mode and load the 16-bit reference register with the time interval that you want the interrupt to occur. There is very little CPU overhead for this timer function, because once the timer is initialized it will run on its own.

Input Capture Function



COUNTER (TCN)

16-BIT RUNNING COUNTER

TCR

CE0-CE1

TINx

Edge Select
& Detect

Latch

16-BIT CAPTURE REGISTER

CAP

Status Flag is set
upon capture
Software should clear by
writing one to CAP bit

Interrupt
CPU

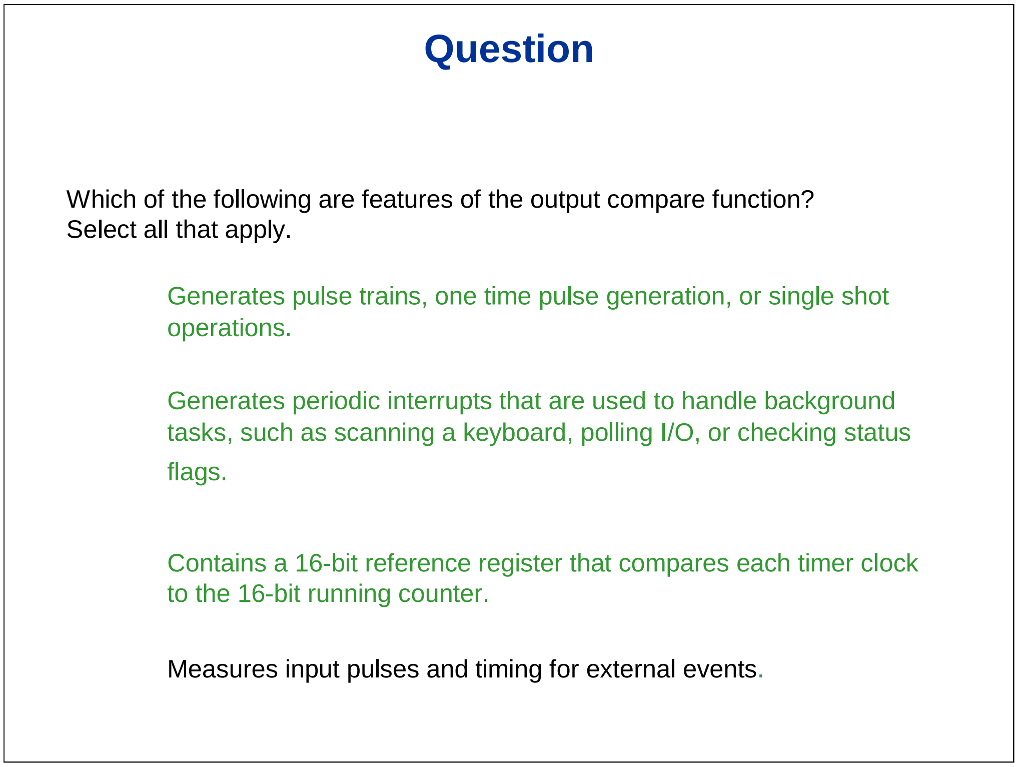
We have examined the output compare function, now let’s take a look at the input capture function.

The input capture function provides a mechanism to capture the time at which an external event occurs. It is very useful for measuring input pulses or timing an external event. The timer has a 16-bit capture register that latches to the counter value when the corresponding input capture edge detector senses a defined transition of “T in”.

The capture edge [(CE)] bits (CE0-CE1) in the time mode register selects the type of transition triggering the capture. The capture choices are based on a rising edge, falling edge, any edge, or to inhibit the T in pin.

Upon a capture event, the status bit CAP will be set and an interrupt will occur if enabled. Accurate time measurements of external events can be measured utilizing input capture. This can occur because it captures or time stamps when the event occurs within one timer clock tick without CPU intervention. The CPU only needs to service the timer before the second event occurs. CPU latency up to this point is irrelevant.

The Tin pin may also be used as an external edge programmable interrupt pin. You only need to set the Tin pin to capture the edge you want and to generate an interrupt when the edge occurs. The contents of the 16-bit capture register is usually ignored.



Question

Which of the following are features of the output compare function?
Select all that apply.

Generates pulse trains, one time pulse generation, or single shot operations.

Generates periodic interrupts that are used to handle background tasks, such as scanning a keyboard, polling I/O, or checking status flags.

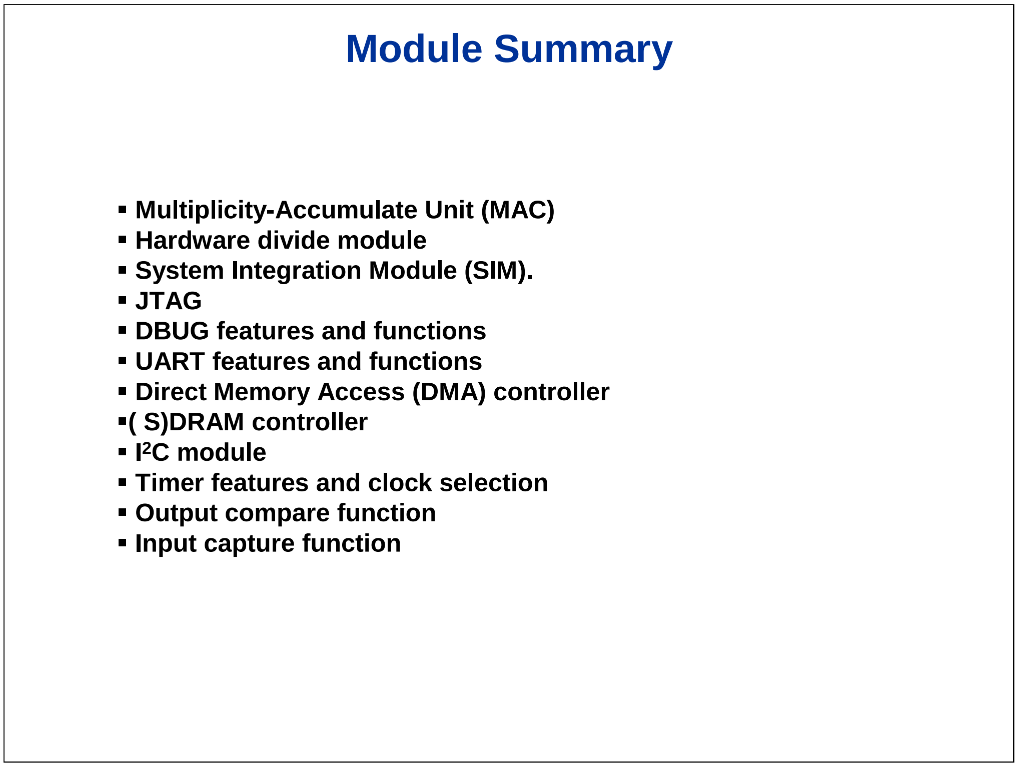
Contains a 16-bit reference register that compares each timer clock to the 16-bit running counter.

Measures input pulses and timing for external events.

Consider this question about the output compare function.

Answer:

The output compare function generates pulse trains, one time pulse generation, or single shot operations. It also generates periodic interrupts that are used to handle background tasks, such as scanning a keyboard, polling I/O, or checking status flags. Finally, it contains a 16-bit reference register that compares each timer clock to the 16-bit running counter.



Module Summary

- Multiplicity-Accumulate Unit (MAC)
- Hardware divide module
- System Integration Module (SIM).
- JTAG
- DBUG features and functions
- UART features and functions
- Direct Memory Access (DMA) controller
- (S)DRAM controller
- I^{2}C module
- Timer features and clock selection
- Output compare function
- Input capture function

Let’s review the ColdFire General Purpose Peripherals that we examined in this module. First, we learned about the features and operation of the of the MAC programming model, the hardware divide module, and system integration. We then looked at the attributes and functionality of JTAG, dBUG, and UART. Next, we examined the features and operations of several controllers including the Direct Memory Access (DMA) and (S)DRAM controllers. Next, we reviewed the I^{2}C module features and operations and Timer features and clock selection. Finally, we learned about the attributes and operations of the timer’s output compare function and input capture function.