Course Introduction

Purpose

• The intent of this course is to provide an overview of the HCS08 microcontroller family, which is part of Freescale’s total microcontroller solutions.

Objectives

• Describe the HCS08’s role in the Freescale 8-bit MCU family.
• Identify the features and advantages of the HCS08 CPU.
• Identify the features and advantages of the following peripheral modules and devices in the S08: Flash, ICG, ICS, SCI, SPI, I²C, TPM, ADC10, New ADC, ACMP, Parallel I/O, and KBI.
• List the S08 security and protection features.
• List the S08 low voltage protection features.

Content

• 24 pages
• 4 questions

Learning Time

• 35 minutes

This course is a general overview of the HCS08 microcontroller family, part of Freescale’s total microcontroller solutions. You will examine how the HCS08’s evolutionary advancements complement microcontrollers that already exist and expand features and performance into a higher range than the HC08.

After looking at the general role of the HCS08 (often referred to as the S08), this course examines the CPU; peripherals and devices such as Flash, the internal clock generator (ICG) module, and the internal clock source (ICS) module; security features and protection; and low voltage protection features.
The S08 fits into the highest level of the Freescale 8-bit family, which originated with the HC05 family. The S08 seamlessly bridges from the HC08 family into the S12 family and beyond. In addition, Freescale offers a full range of other products and tools that are compatible with the S08 family as well as the HC08 and S12 families.
Why HCS08?

• Evolutionary step in the popular HC08 family
• Extended battery life technology
• High-level performance without compromising low power
• Flash leadership with the third generation of 0.25µ Flash technology
• Innovative on-chip debugger, which speeds up your time to market
• SoC methodology:
  – New product derivatives
  – High level of integration also reduces system costs
• Robust EMC performance
• 100 percent object code compatible with HC08

Here you can see an example of an S08 controller: the MC9S08GB60. The S08 is an evolutionary step in the popular HC08 family; it's not meant to be revolutionary. Its purpose is to complement microcontrollers that already exist and to expand features and performance into a higher range than where the HC08 was originally targeted. To achieve this purpose, the S08 has technology to extend battery life, low power enhancements, and increased performance at low voltage. As a result, you get high-level performance without compromising low power even from battery-powered applications. With other chips, you might not see this level of performance at lower voltages.

The S08 extends Freescale’s Flash leadership with the third generation of 0.25µ Flash technology. This technology is one of the fastest programming, densest arrays available in the 8-bit MCU market.

The S08 also introduces an innovative on-chip debugger, which speeds up your time to market. This system eliminates the traditional need for a separate hardware emulator, and it allows you to emulate your systems using the actual silicon you will use in your final applications.

We also have introduced system-on-a-chip (SoC) methodology. This synthesized design methodology allows you to rapidly integrate new products that have different modules and different feature sets; therefore, you can expand the S08 family by rapidly producing new product derivatives with a high level of quality. Imagine: you can reuse similar designs from previous products, and you can bring together a variety of peripherals on your application board! SoC’s high level of integration also reduces system cost.

The S08 also has robust EMC performance. Several enhancements improve the chip’s performance in noisy environments; EMC susceptibility has been reduced and minimized. It also has new features that make it easier to reduce and control the radiated emissions from the chip so that this MCU does not affect other parts of your system.

Finally, as part of the goal to make the S08 an evolutionary step, it is 100 percent backwards code compatible with HC08, but it also has some expanded instructions and some addressing modes to improve C code generation efficiency.
Here you can see one of our typical general-purpose S08 MCUs: the MC9S08QG8. You will see this diagram throughout the course used as an example of where peripherals and devices can be found in S08 MCUs. This MCU has a variety of peripherals, and this is just a sample of the overall module library that Freescale has available for this technology.

On the left side of the block diagram, you can see several of the systems that are very common throughout the whole S08 family. The CPU core has an integrated on-chip in-circuit emulator (ICE) module, which allows the emulation of the MCU without additional hardware. Also, many of the system control functions are grouped together in a single piece of logic for easy interaction with the CPU. It's separated from the MCU to allow for flexible resets, interrupts, and different MCU-type system-level functions without making modifications to the CPU core. As a result, the CPU is very robust.

Another feature of all S08s is an on-chip voltage regulator. This voltage regulator supplies a regulated voltage to all the digital logic because all S08 products are manufactured with 0.25 µm technology. This technology has a 2.7V maximum $V_{DD}$ limit, so a voltage regulator is required to apply voltages above 2.7V. The voltage regulator is advantageous because you can run the MCU across a wide range of voltages while keeping the voltage regulated at a single voltage—you see very few voltage-induced performance tradeoffs.

Analog voltage supplies may be provided by external pins or tied on-chip to $V_{DD}$.

Port pins receive power straight from $V_{DD}$, which allows input and output signals to go to the voltage rails.
**S08 CPU**

- High performance HCS08 CPU:
  - 50 ns minimum instruction cycle time down to 2.1V with a 20 MHz bus
  - 125 ns minimum instruction cycle time down to 1.8V with a 8 MHz bus
- Object code compatible with HC08s but also has added instructions and expanded addressing modes:
  - 10 to 15 percent increase in C code compiled efficiency
  - Single-byte Multiply and Divide instructions
- New addressing modes for the LDHX instruction:
  - EXT
  - IX
  - IX1
  - IX2
  - SP1
- New addressing modes for STHX and CPHX instructions:
  - EXT
  - SP1
- New BGND instruction
- Instruction queue (or pipeline)

The S08 CPU is a performance enhancement over the HC08 CPU. The high performance HCS08 has a 50 ns minimum instruction cycle time down to 2.1V with a 20 MHz bus and 125 ns minimum instruction cycle time down to 1.8V with a 8 MHz bus. It is one of the fastest 8-bit MCUs in high volume production.

The S08 CPU is object code compatible with HC08s, but it also has added some instructions and expanded addressing modes to improve code efficiency. In general, you should see a 10 to 15 percent increase in C code compiled efficiency. Also, there are single-byte Multiply and Divide instructions.

Several new addressing modes have been added for the Load HX (LDHX) instruction, including extended addressing mode (EXT), indexed—no offset (IX), indexed—8-bit offset (IX1), indexed—16-bit offset (IX2), and stack pointer relative—8-bit offset (SP1).

New addressing modes for the Store HX (STHX) and Compare HX (CPHX) instructions have also been added: EXT and SP1. These extended addressing modes or additional addressing modes really help C code to efficiently compile and handle 16-bit numbers.

Also introduced in the S08 is the background (BGND) instruction. It basically acts as a software breakpoint when you are using the on-chip ICE to debug your system software, allowing the user to add as many breakpoints as needed.

Finally, Freescale has also introduced a new instruction queue or pipeline that improves the overall flow of the CPU. This improves performance and allows faster execution speeds.
Question

The purpose of the HCS08 is to complement microcontrollers that already exist and to expand features and performance into a higher range than where the HC08 was originally targeted. How does it accomplish this task? Select all that apply and then click Done.

- Extended battery life technology
- New, more efficient object code
- Low power enhancements
- Increased performance at all voltages
- Revolutionary approach to CPU design

Please select all the statements that are correct concerning the purpose of the HCS08.

Correct.

To achieve its purpose, the S08 has extended battery life technology, low power enhancements, and increased performance at all voltage. The object code is not all new; it is 100 percent object code compatible with HC08 with some expanded instructions and addressing modes to improve C code generation efficiency. The CPU’s design is enhanced, but not revolutionary since the purpose is to complement existing MCUs rather than create a new family.
The S08 utilizes Freescale’s third-generation 0.25µ Flash technology, which makes Flash more affordable because its size is approaching the size of ROM modules. It is one of the most dense Flash arrays used and embedded onto an MCU.

It offers 100,000 write/erase cycles typical at 25°C and 10,000 minimum from -40°C to +85°C. This allows many iterations of reprogrammability, and it also allows use of the Flash array as a data-type e-squared within your system.

This Flash also has very robust data retention. It has a 15-year minimum from -40°C to +85°C and more typically, a 100-year lifetime at 25°C.

The Flash is fully operational, both reads and writes, down to 1.8V without any external voltage other than MCU supply. This allows the Flash to be fully utilized across the entire operating voltage of the MCU.

Self-timed programming makes programming the Flash simple, quick, and easy. The Flash has a Flash controller interface that controls the timing of the Flash programming and erasing. This module makes it easier to write code for Flash e-squared type applications, when you want to change the Flash content within your application. Since this control module controls the timing for the actual programs and erases, the user does not need to know the exact timing or come up with timing-related delays in order to program or erase Flash.

The Flash module also programs and erases very rapidly; you can program 8 bits in 20 µs, thereby reducing production programming costs and power consumed while programming. Finally, a 512-byte page can be erased in 20 ms.

In-application re-programming makes field upgrades easy.

Freescale has also added a very flexible block protection scheme and enhanced security features over the HC08. Let’s sum it up. Using Freescale’s Flash is simple and straight forward!
Here is the block diagram for the internal clock generation module or ICG for short. The ICG generates the clocks used by the CPU and chip peripherals. The ICG is composed of Frequency-locked loop or FLL for short and the clock selection circuitry. There are two clock references available, an external clock input and oscillator, and a very accurate internal reference generator.

The ICG module provides all the system clocks to all the modules and to the CPU of the device. This particular module has four modes of operation: Self-clock mode (SCM), FLL engaged, internal clock (FEI) mode, FLL bypassed, external clock (FBE) mode, and FLL engaged, external clock mode (FEE).

First, SCM is a simple, internally clocked mode. It’s the default mode out of reset, and it provides a very reliable start-up from a power-off situation, or for wake-up from some Stop modes. The default frequency of ICGOUT is approximately 8 MHz. This mode, although it provides a very reliable clock source, is not the most accurate option available.

Second, the FEI mode is more accurate than the SCM. The FEI mode uses an internal clock reference that can be trimmed ± 25 percent to within 0.4 percent accuracy. Once trimmed, its typical deviation across temperature and voltage is only ± 0.5 percent. With the FLL engaged, there are eight different multipliers to increase your frequency (ICGOUT is 8.8 MHz to 40 MHz), plus another eight different divides that can be used to reduce the frequency back down to as low as 64 KHz, allowing for many combinations of frequency output.

The third mode is FBE. This mode simply takes an external clock source presented at the MCU pins and uses that clock source directly to generate the system clocks. In this case, the bus clock is one-half of the external clock frequency; ICGOUT is DC to 40 MHz.

The fourth mode, FEE, is very similar to the FEI mode except instead of using an internal reference for the FLL, it uses an external clock reference. This external clock can be either a crystal or an external system-level clock that runs across the board. This mode has eight different multipliers, and ICGOUT is 62.5 KHz to 40 MHz.

Two of these four modes (SCM and FEI) require no external components, which reduces your system's cost and increases its reliability by not having additional components on your board. All four modes are software selectable; the user program can switch between modes as necessary within the system. This feature allows a great deal of flexibility within your system for clocking, providing accurate clocks when necessary, and providing very reliable start-up clocks at other times when accuracy may not be as important.

The external oscillator option can be used in FBE or FEE modes, and there are low power or high gain external oscillator options. The low power option limits the voltage swing on the pins when an external crystal oscillator is used, minimizing the power consumption of the oscillator. The high gain option drives the oscillator pins to rail for a more robust option in noisy environments. However, driving the pins harder results in higher power consumption.

The ICG module also has a clock monitor that automatically switches modes if a clock is lost. The clock monitor can be used in conjunction with the external oscillator in the FEE or FBE modes to monitor and make sure that an external clock is always available. If the external clock deviates outside of an expected range, either a programmable interrupt or a reset will be generated. At this point, the system can recover and either try to restart the external oscillator or optionally go into an alarm state or safe state. It can also be disabled for power savings.

All four modes have the ability to use a software selectable reduced frequency divider (RFD) that is available within the ICG module. This RFD has eight different divide options (divide by 1, 2, 4, 8, 16, 32, 64, or 128). When used in one of the FLL engaged modes, it provides rapid changes of frequency without having to wait for the FLL to reacquire a lock for accurate time measurement.

Finally, the output of the ICG is divided by two to create the bus frequency as seen in this equation. Click the “ICG” button in the upper right hand corner of the page for a summary of these ICG module features.
ICG

• Four modes of operation:
  – SCM - Default mode out of reset: ICGOUT is roughly 4 MHz
  – FEI mode - 8 different multipliers: ICGOUT is 64 kHz to 40 MHz
    • Reference trimmable ± 25% to within 0.4% accuracy
    • Reference accurate to typical +/- 0.5% across temperature and voltage
  – FBE mode - bus = ½ ext clock: ICGOUT is DC to 40 MHz
  – FEE mode - eight different multipliers: ICGOUT is 62.5 KHz to 40 MHz
• Two modes require no external components
• All modes are software selectable; user program can switch between modes at any time
• Low power or high gain external oscillator options:
  – Low power limits voltage swing on oscillator pins to minimize power consumption
  – High gain drives oscillator pins to rail for robust operation in noisy environments
• Clock monitor:
  – Automatically switches modes if a clock is lost
  – Software selectable for reset or interrupt on lost clock
  – Can be disabled for power savings
• Software selectable RFD available in all ICG modes:
  – Divide by 1, 2, 4, 8, 16, 32, 64, or 128
  – Allows frequency changes without losing FLL Lock
• Bus frequency = ½ ICGOUT

Reference material for previous page
Here is the block diagram for the internal clock source module, or ICS for short. The ICS is very similar to the ICG module used on some other HCS08s. Like the ICG, the ICS is composed of FLL and the clock selection circuitry. There are two clock references available, an external clock input and oscillator, and a very accurate internal reference generator.

The ICS module has four modes of operation, three of which are basically the same as the ICG: FEI mode, a new FLL bypassed, internal clock (FBI) mode, FBE mode, and FEE mode.

The first mode is FEI mode. The FEI mode enables the FLL. The internal reference, which is set between 31 and 39 KHz, is the source for the loop. The loop is preprogrammed to set to a single multiply of 512, resulting in a final ICSOUT frequency of 16 MHz to 20 MHz.

FEI is the default out of reset. Since the internal reference is not trimmed after reset, the output (ICSOUT) is roughly 8 MHz. A more accurate trim value can be written to the ICS by user code at any time after reset.

The next mode is the FBI mode. This mode relies on the internal clock reference directly. This reference can be programmed to be between 31.25 KHz and 39.06 KHz. Since the internal reference can be trimmed, FBI is an accurate clock mode requiring no external components.

Next is the FBE mode. The ICSOUT comes directly from the external clock or external oscillator and can be anywhere in the range of DC to 20 MHz. ICS FBE mode is the same as the ICG FBE mode.

The FEE mode is basically the same as the FEI mode except it uses an external clock reference instead of the internal clock reference. The loop is still preprogrammed for a single multiply value of 512, so it has the same ICSOUT frequency outputs as the FEI mode, which is 16 MHz to 20 MHz.

Of the four modes, two modes (FEI, and FBI) require no external components, thereby increasing the system’s reliability and freeing up board space for other use.

The internal reference of the ICS is trimmable, with a typical resolution of 0.1 percent. Deviation, once trimmed, is typically about +0.5 percent to -1 percent.

All of these modes are software selectable. The software can select between any of the four modes at any time, again allowing for very flexible clock applications to switch between high frequency, low frequency, and internal or external as needed by the application.

The oscillator portion of the ICS has the same low power or high gain oscillator options as the ICG’s oscillator. Low power limits voltage swing on the oscillator pins to minimize power consumption, and high gain drives the oscillator pins to rail for robust operation in noisy environments.

The ICS also has a software selectable bus divider, BDIV, similar to the RFD on the ICG, and it is available in all ICS modes. It has four different divide options: 1, 2, 4, or 8. It also allows frequency changes without losing the FLL Lock. The BDIV defaults to divide by 2 after any reset.

Finally, like the ICG, the bus frequency is one-half of the ICSOUT frequency as seen in this equation. Click the “ICS” button in the upper right hand corner of the page for a summary of the ICS module features.
ICS

• Four modes of operation:
  – FEI mode - ICSOUT is internal clock x 512: 16 MHz to 20 MHz
  – FBI mode - ICSOUT is internal clock directly: 31.25 KHz to 39.06 KHz
  – FBE mode - ICSOUT is external clock directly: DC to 20 MHz
  – FEE mode - ICSOUT is external clock divided down x 512: 16 MHz to 20 MHz

• Two modes require no external components

• Internal reference is trimmable with typical resolution of 0.1%
  – Deviation from trimmed frequency is typically +0.5% to -1%

• All modes are software selectable, FLL open loop is not
  – User program can switch between modes at any time

• Low power or high gain oscillator options
  – Low power limits voltage swing on oscillator pins to minimize power consumption
  – High gain drives oscillator pins to rail for robust operation in noisy environments

• Software selectable bus divider, BDIV, available in all ICS modes
  – Divide by 1, 2, 4, or 8
  – Allows frequency changes without losing FLL Lock
  – Defaults to divide by 2 after any reset

• Bus frequency = ½ ICSOUT

Reference material for previous page
Here you can see the serial communications interface (SCI) module in the MC9S08QG8 block diagram. It provides a basic two-wire, asynchronous communications interface that can connect to a PC, other peripherals, or controllers in an application. The maximum baud rate is 1.25 Mbps, with a maximum bus frequency of 20 MHz for most of the S08 family.

The SCI module has a flexible, independent 13-bit modulo-based baud rate generator, which allows a wide variety of baud rates to match the bus frequency to external communication standards accurately.

Advanced data sampling at 16x over sampling, which provides very reliable communications. When communicating through a cable to another board several feet away, you will still see reliable communications.

Both the receive and transmit are double buffered. Double buffering allows the SCI to queue up an extra byte for transmit. Upon reception, it also allows the flexibility to wait until reading the data.

The SCI also has a hardware parity and receiver wakeup to align the SCI with standard communications protocols on the market, and there is an optional 13-bit break available on some of the newer S08 MCUs.
Let’s examine the serial peripheral interface (SPI) module, which is a four-pin synchronous serial communications interface. It allows up to 10 Mbps in master mode or up to 5 Mbps in slave mode based upon a maximum bus speed of 20 MHz.

It has a double-buffered receive and transmit to allow more flexibility for queuing messages or receiving messages.

The master mode and slave mode with slave select output are software selectable. It has a full-duplex mode with separate output and input lines or a single-wire, bi-directional mode that operates in a half-duplex mode.

The SPI has software selectable clock phase and polarity options. The clock polarity option inserts an inverter in series with the clock. The phase option chooses between two different phases between the clock and data. Also through software, the data is selectable to be most significant bit (MSB) or least significant bit (LSB) first.
Can you remember the four modes of operation for the ICS module? Match each mode on the left with its description on the right. Click “Done” when you are finished.

A  FEI
B  FBI
C  FBE
D  FEE

A  Enables the FLL; internal reference as the source for the loop; ICSOUT 16 MHz to 20 MHz
B  Relies on internal clock reference directly; ICSOUT 31.25 KHz to 39.06 KHz.
C  ICSOUT comes directly from external clock; ICSOUT 20 MHz
D  Uses external clock reference; loop preprogrammed for a single multiply value of 512; ICSOUT 16 MHz to 20 MHz

Let’s take a moment to review the ICS modes.

Correct.

FEI mode enables the FLL, and the internal reference is the source for the loop; ICSOUT is 16 MHz to 20 MHz. FBI mode relies on an internal clock reference directly; ICSOUT is 31.25 KHz to 39.06 KHz. FBE mode’s ICSOUT comes directly from the external clock; ICSOUT is DC to 20 MHz. FEE mode uses an external clock reference, and it is loop preprogrammed for a single multiply value of 512; the ICSOUT is 16 MHz to 20 MHz.
The inter-integrated circuit bus (I²C) module can run up to 100 Kbps with maximum bus loading, or it can also run up to 1 Mbps with reduced bus loading, at a maximum bus rate of 20 MHz.

Open-drain drivers allow for multi-master operation.

The clocking schemes are very flexible and software programmable for one of 64 different serial clock frequencies.

The module also has a software selectable acknowledge bit.

The I²C module allows for an interrupt-driven, byte-by-byte data transfer, as opposed to a flag-driven data transfer, for reduced CPU overhead during communications.

Automatic mode has an arbitration lost interrupt for switching from master mode to slave mode. It helps facilitate multi-master operation.

The I²C module also has a calling address identification interrupt that allows the MCU to ignore message traffic on the I²C bus except when it's directly addressed.

The I²C also supports START and STOP signal generation/detection, repeated START signal generation, acknowledge bit generation/detection, and bus busy detection.
The Timer/PWM module (TPM) is composed of a 16-bit timer with zero to eight programmable channels depending upon the module implementation. Each channel can be configured to be an input capture, output compare, or PWM output. The input capture can be rising-, falling-, or any-edge input capture trigger. On output compare, the channel can be configured to set, clear, or toggle upon output compare.

Each and every channel can be configured for full-buffered PWMs, either center-aligned or edge-aligned. There can be potential noise issues with multiple edges on the same clock cycle; center-aligned configuration has no aligned edges for reduced noise. The PWM outputs have selectable polarity. These features provide a very flexible modulator output for applications, such as simple motor control.

The clock source for each TPM is independently selectable as a bus clock, a fixed system clock, or an external pin, also called a TPMCLK. Each of these clock sources will input into the TPM’s prescaler, which has eight software selectable prescale choices: 1, 2, 4, 8, 16, 32, 64, or 128. When selected, the fixed system clock (XCLK) and TPMCLK are synchronized to the bus clock. On most S08 MCUs, XCLK is the external clock, so if an oscillator or external clock is available on the board, that can be used directly, even if the selected clock mode is using the FLL loop. The external clock pin input can run up to the bus frequency divided-by 4.

The timer itself is 16-bit free running or, in the case of the center-aligned PWM, up/down count operation.

The TPM also has a 16-bit modulus register to control the counter range. It can be used to select any value from one to FFFF as the match value to set the overall period and time-out of the timer.

There is one interrupt per channel, plus a terminal count interrupt. The terminal count interrupt is set at the modulus time out value. The channel interrupt can be set for the input compare, PWM or output compare functions. Click “TPM Summary” to review the TPM’s features.
TPM Summary

• 16-bit timers with programmable channels:
  – Each channel may be input capture, output compare, or PWM
  – Rising-edge, falling-edge, or any-edge input capture trigger
  – Set, clear, or toggle output compare action

• Each and every channel can be configured for full buffered PWMs
  – Configurable for center-aligned or edge-aligned PWMs
  – Selectable polarity on PWM outputs

• Clock source to prescaler for each TPM is independently selectable
  as bus clock, fixed system clock, or an external pin:
  – Prescale taps for divide by 1, 2, 4, 8, 16, 32, 64, or 128
  – XCLK and pin paths are synchronized
  – External TCLK pin running up to f_{BUS}/4

• 16-bit free-running or up/down (CPWM) count operation
• 16-bit modulus register to control counter range
• One interrupt per channel plus terminal count interrupt

Reference material for previous page
The S08 family has two versions of 10-bit Analog-to-digital converters, the original one found on the MC9S08GB/GT family, referred to as the ATD and a newer version introduced on the MC9S08QG family referred to as the ADC10.

Let's discuss the ATD version first.

It is fully functional from 1.8V to 3.6V, and at 1.8V, it has exceptional performance when compared to the competition, with total error typically of 1.1 LSB and maximum error only 2.5 LSB. This error range includes inherent quantization error (1/2 LSB) and circuit error (differential, integral, zero-scale, and full-scale).

Single conversion times for a ten-bit value are 14 µs, with a conversion frequency of 2 MHz.

Ten-bit values can be either left- or right-justified in unsigned data mode, and they are left-justified in signed data mode.

The ADC module includes conversion complete flags and conversion complete interrupt generation.

Each ADC module can have up to eight analog input channels, and the module has an analog input multiplexer for them. Typically, each analog input is also MUXed with other MCU pin features.

The ADC supports either a single or a continuous conversion mode, depending on whether multiple conversions are required or not. The ADC will automatically enter a low power mode when the MCU enters Stop mode. This low power mode can also be set through ADC register bits.
Freescale has recently introduced a new ADC module with new features for its S08 family, such as the MC9S08QG8. This ADC has faster conversion times—2.5 µs for a 10-bit single conversion time at an ADC frequency of 8 MHz (versus 14 µs for a ten-bit conversion with the old module).

The new ADC has its own asynchronous ADC clock (ADCCLK), which allows the ADC to operate in Stop 3 (low power) mode where system clocks are not normally available.

It also has a programmable compare function, which allows the user to set a specific value in a compare register and then generate an interrupt on values that are either less than or greater than/equal to the compare value. The 10-bit match value is software programmable.

The new ADC also has an optional hardware conversion trigger. Typically, this hardware conversion trigger is connected up to a real-time interrupt (RTI) module on the S08, which allows for a selectable periodic conversion to continuously occur. In continuous conversion mode, the conversions occur as fast as every 2.5 µs. Using the RTI to generate periodic conversions, the conversions occur less frequently thereby consuming less power.

An on-chip temperature sensor is selectable on an internal ADC channel. This sensor can be used to calculate relative temperature changes over time or when calibrated in the application, can be used to determine actual operating temperature.

An on-chip constant voltage band-gap reference can be configured to be the input to an internal ADC channel. The conversion value on this constant voltage can be used to determine the actual voltage reference for the ADC. This is useful in battery-powered systems where the external voltage reference will drop over time.

The new ADC is software configurable for long or short sample times and high speed or low power operation. The modes can be mixed or matched depending on the system requirements, which allows flexible configuration to meet the needs of your application.

A couple of the original ADC features have been eliminated in the new ADC. The left-/right-justified result data is no longer an option; ten-bit values are always right-justified. The left-justified signed data mode has also been eliminated. Left-justifying and converting to a signed value can easily be accomplished in software.
Next, the analog comparator (ACMP) module operates from a full rail-to-rail supply. There is less than 40 mV of input offset and less than 15 mV of hysteresis to provide reliable and consistent triggering. The comparator has a selectable interrupt on rising edge, falling edge, or both edges on the comparator output.

The positive input can optionally use an internal bandgap, which frees an I/O pin for other uses. This also provides an accurate constant voltage source as a reference voltage.

There are two options for comparator output. The first option is to make the comparator output visible on an external pin so that other pieces of the application can see when the comparator triggers. The second option is to internally route the comparator output to a TPM channel for input capture.
Parallel I/O

• Individually programmable pull-ups on each pin
  – Reduce system cost and increase reliability by eliminating components

• Software-controlled slew rate output buffers
  – Slower rate can reduce EMC emissions
  – Selectable pin by pin

• Software controlled drive strength on newer MCUs
  – Typical high drive current ≥ 10 mA
  – Typical low drive current ≥ 2 mA
  – Selectable pin by pin

The parallel I/O on the S08 systems all have programmable pull-ups on each pin. Each individual pin within a specific port can be selected to have the pull-up enabled. The pull-ups reduce system cost and increase reliability by eliminating external components that take up board space, take up budget, and provide potential for breakage and error. The default option after a reset is I/O pull-ups disabled.

The software-controlled slew rate output buffers are another option for each I/O pin. Each output pin can be controlled to have either a standard slew rate or a slower slew rate. The slower slew rate can reduce EMC emissions from a fast-changing I/O pin. The slew rate is selectable pin by pin and the default out of reset is the standard slew rate.

Software controlled drive strength on newer MCUs can be set up with typical high drive current of at least 10 mA, which is very useful for lighting LEDs and even some infrared LEDs. Typical low drive current is a minimum of 2 mA. By limiting the drive strength, you can reduce your EMC emissions by not driving external signals as hard and generating as much noise. The drive strength is also selectable pin by pin. After a reset, drive strengths are set to the low drive option.
Let's examine the keyboard interrupt (KBI) module. Each one can have one to eight associated input pins. Each pin is individually software selectable as an interrupt; therefore, if you have a port of eight pins, you can pick pins one and five, or whatever configuration meets your system requirements.

There is a single interrupt vector for each KBI module. For example, if a single KBI module with eight pins is available on the MCU, all eight pins will share the same interrupt vector. If there are two KBI modules with eight pins apiece, each group with eight pins will have its own interrupt vector.

Each KBI module is software selectable as active high or active low, and this can be chosen pin for pin on each module. The KBI module is also software selectable as edge only or edge and level interrupts. This feature is selected for all pins within one KBI module.

The port I/O pull-up enable register is also used to enable pull-ups and pull-downs for the KBI function. On a pin-by-pin basis, they can be enabled or disabled as required by the system. The corresponding port data register can be read to determine which pins triggered, and this data can be read without disabling the KBI functionality.

All KBI module configurations can be modified at any time by software, including all the features just discussed, which make the KBI module very useful for interrupts that may need to change functionality over time.
The S08 has several security and system protection features. For enhanced security, two NVM bits have been reserved in a special register. If the security is enabled, then an 8-byte security code is required to access and read the Flash contents. This feature prevents a hacker from trying to get into an MCU to read out the code or any sort of data contents that are within the Flash. If security is not broken, the only Flash function that can be executed is a mass erase.

With regards to system protection features on S08 MCUs, a Computer Operating Properly (COP) watchdog is available, and it has a periodic timeout if the COP is not fed by writing to a specific register within the timeout period. The COP can operate off of the bus clock or an internal clock source independent of the bus clock. When using the internal clock source, the COP can time out if the bus clock is lost and the CPU is not operating properly at the time.

Another protection feature is illegal address and illegal opcode detect/reset. If an unimplemented address or unimplemented opcode attempts to execute, the system will reset.

The Flash also has a protection feature that prevents unintentional changes to Flash memory. The Flash block protection settings are loaded at reset and cannot be modified by the user code. The settings are programmable such that either the entire array or just portions of the array can be protected.
Let's review security and protection features.

Correct.

If the security is enabled, then an 8-byte security code is required to access and read the Flash contents. If security is not broken, the only Flash function that can be executed is a mass erase.
Two features provide low voltage protection. The low voltage detect (LVD) module can be configured as a reset, an interrupt, or a software flag. The nominal trip point is 1.8V; this is the default out of a power on reset (POR). LVD can also be reconfigured to trip at 2.1V as a second software selectable trip point. In addition, this module can be set to function in Stop mode for protection while in the low power state.

The second low voltage protection feature is the low voltage warning (LVW). This additional warning that $V_{DD}$ is falling is set at a higher trip point than the LVD. It is a flag, not an interrupt or a reset, and can be selectable for either 2.1V or 2.4V nominal. This additional low voltage warning flag allows a system to detect when batteries are getting low but still have enough power to put the MCU into a ready state for replacing batteries.
Is the following statement true or false? Click Done when you are finished.

“The LVD will always generate either a reset or an interrupt.”

True
False

Consider this question concerning low voltage protection.

Correct.

The LVD can be configured so that only a software flag is set and no interrupt or reset is generated.
Course Summary

- HCS08 evolutionary role in the Freescale 8-bit MCU family
- Features and advantages of the S08 CPU
- Features and advantages of the peripherals and devices in the S08: Flash, ICG, ICS, SCI, SPI, FC, TPM, ADC10, new ADC, ACMP, parallel I/O, and KBI
- S08 security and protection features
- S08 low voltage protection features

In this course, you learned about the HCS08 microcontroller family, which is part of the Freescale family of controllers. The S08’s evolutionary advancements complement microcontrollers that already exist and expand features and performance into a higher range than the HC08.

After looking at the general role of the HCS08, this course examined the CPU, which is one of the fastest 8-bit MCUs in mass production, but it is still object code compatible with HC08s. You also learned about its peripherals and devices, such as Flash, the SCI module, the SPI module, the TPM module, and the new ADC module. For example, the S08 utilizes Freescale third-generation 0.25µ Flash technology, which makes Flash more affordable because its size approaches the size of ROM modules. It is one of the most dense Flash arrays used and embedded onto an MCU.

Finally, you also learned about security features and protection, and low voltage protection features. The security features protect code and data within the Flash from hackers and the system protection features help prevent the MCU from malfunctioning, even in harsh environments. The low voltage detection and warning features allow for very flexible configurations for protecting the MCU from sudden or gradual voltage loss.