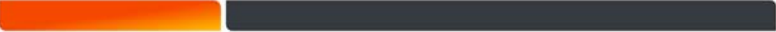






**Agenda** 

1. Module Overview
2. On-chip Interconnections and Inter-module Dependencies
3. Software Configuration
4. Example / Use Case
5. SIM Frequently Asked Questions (FAQs)

 External Use | 1 

In this presentation we'll cover:

- An overview of the system integration module
- The on-chip interconnections and inter-module dependencies
- Software configuration
- An example use case
- And some frequently asked questions



Let's first begin with an overview of the module.

## Module Overview

- The System Integration Module (SIM) provides system control and chip configuration registers
- Functions fall into these main groups:
  - Clock source selection
  - Clock divide values
  - Clock gating control
  - Module configurations
  - Chip configuration information

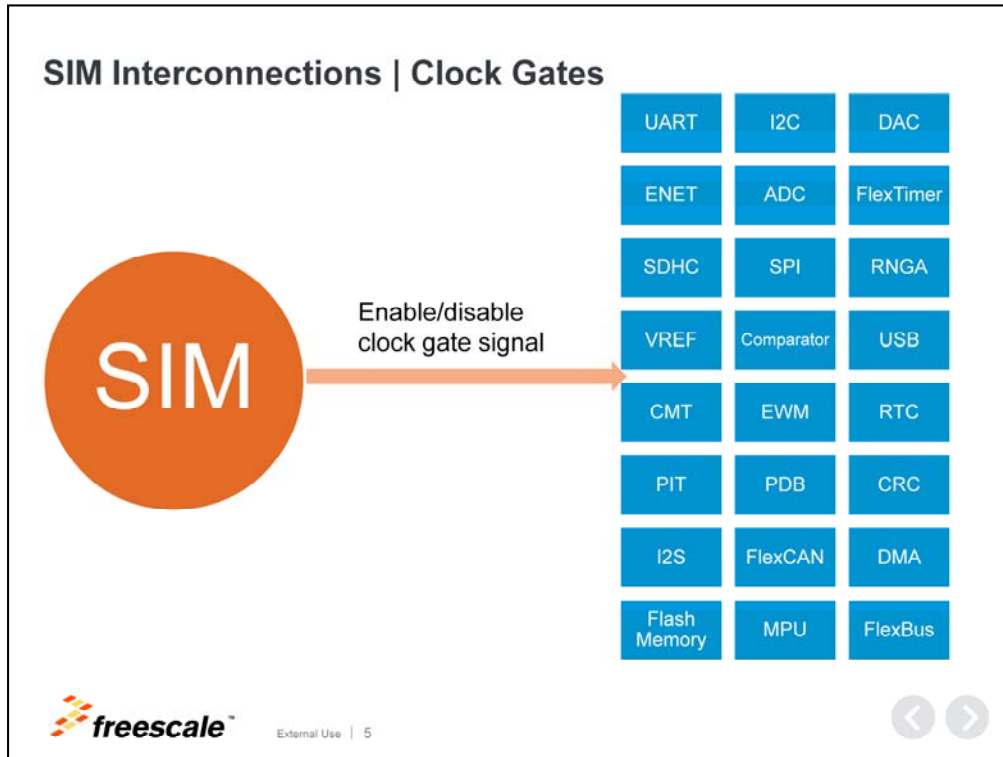
## Module Overview

The System Integration Module, or SIM, is essentially a glue logic block that provides control of system-on-chip inputs to almost all of the modules on the chip.

The main functions of the SIM are clock source selection, clock divide values, clock gating control, module configurations and chip configuration information.

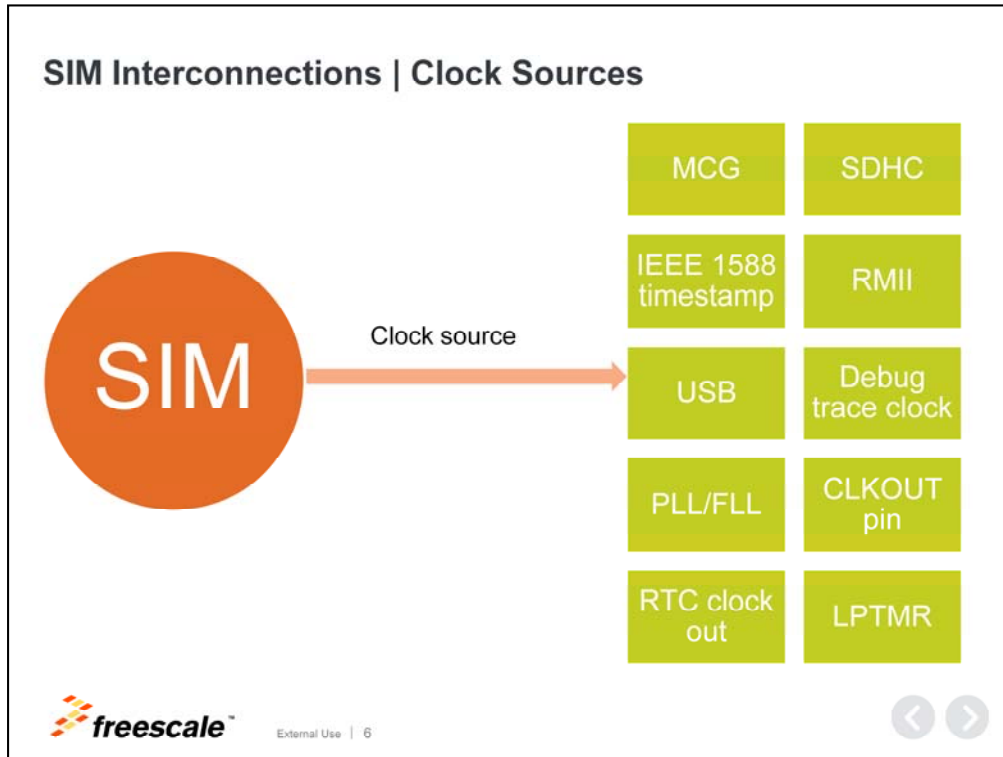
## 2. On-chip Interconnections and Inter-module Dependencies

Now, let's talk about on-chip interconnections and inter-module dependencies.



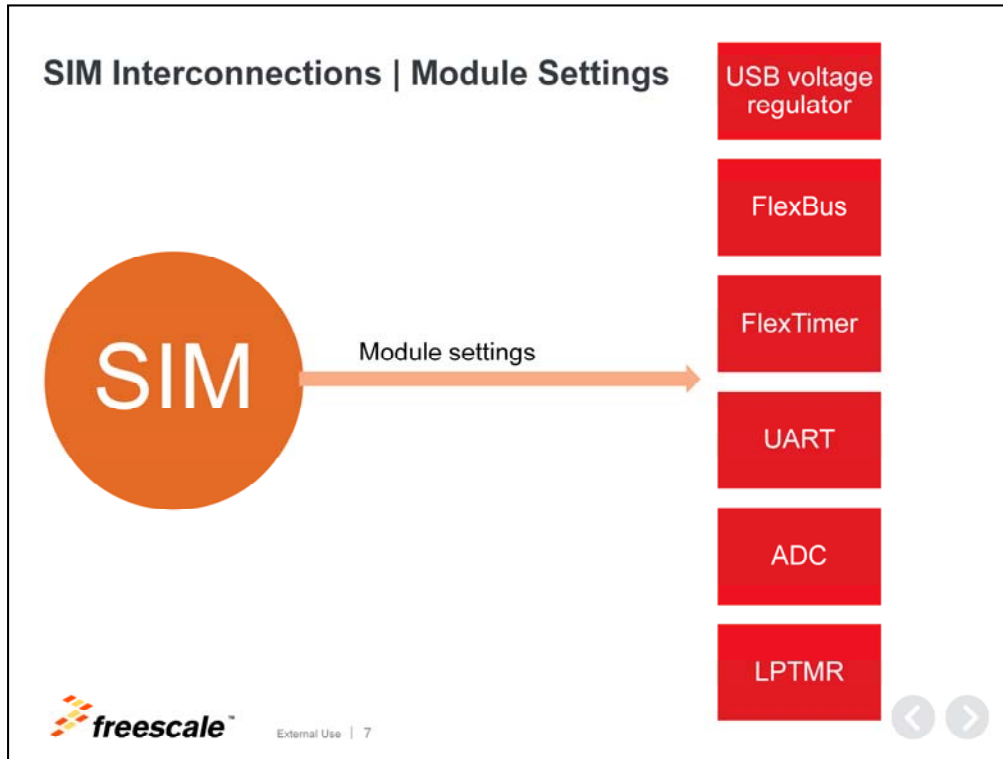
### SIM Interconnections | Clock Gates

The SIM interconnects to almost all modules on the chip, **and controls the clock gates.**



### SIM Interconnections | Clock Sources

The SIM interconnects to almost all modules on the chip, **and selects the clock sources.**



## SIM Interconnections | Module Settings

The SIM interconnects to almost all modules on the chip, **and can configure the settings of many modules.**



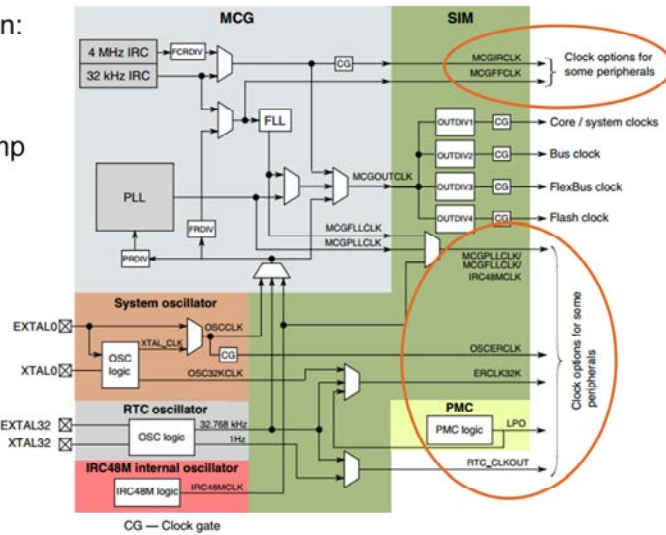


In this next section we'll cover software configuration.

## Clock Source Selection

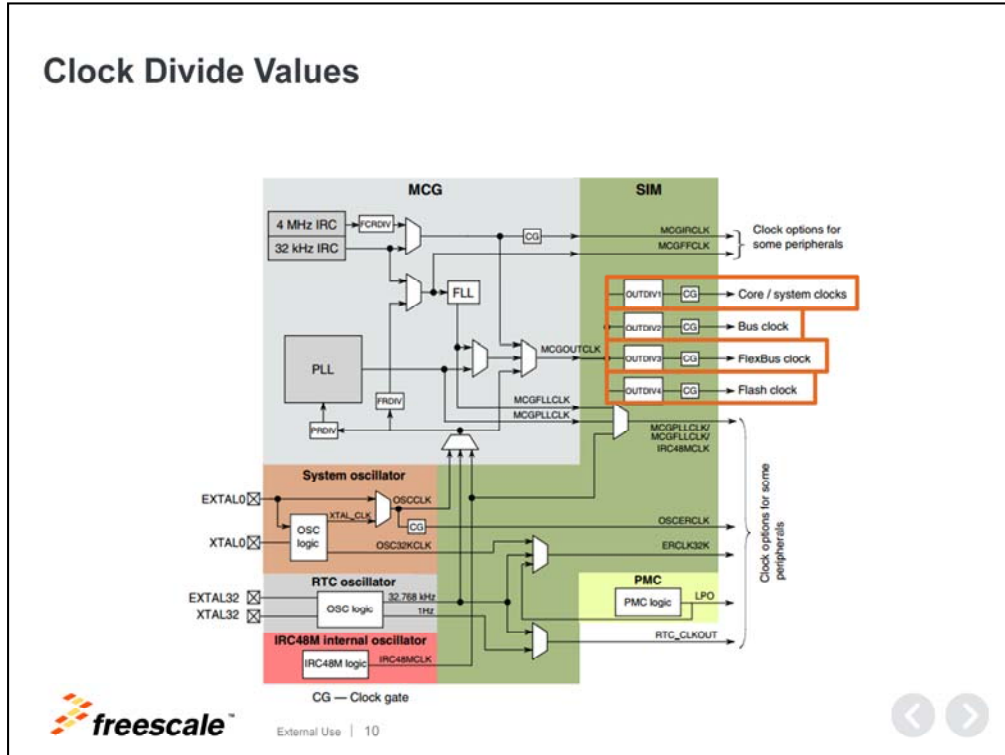
Clock source selection:

- MCG
- SDHC
- IEEE 1588 timestamp
- RMII
- USB
- Debug trace clock
- PLL/FLL
- CLKOUT pin
- RTC clock out
- LPTMR



## Clock Source Selection

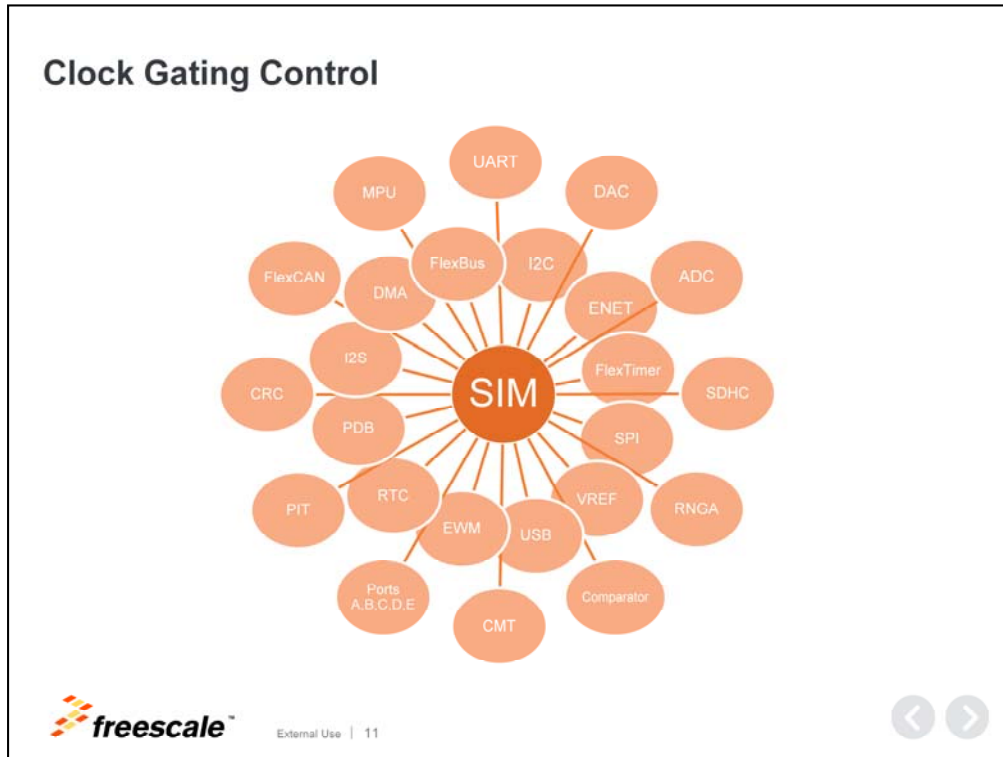
The SIM offers clock source selection for modules that have the option of selecting their clock source, such as the SDHC module, Ethernet timestamp, USB and debug trace clock.



## Clock Divide Values

The SIM can also modify the output dividers of the:

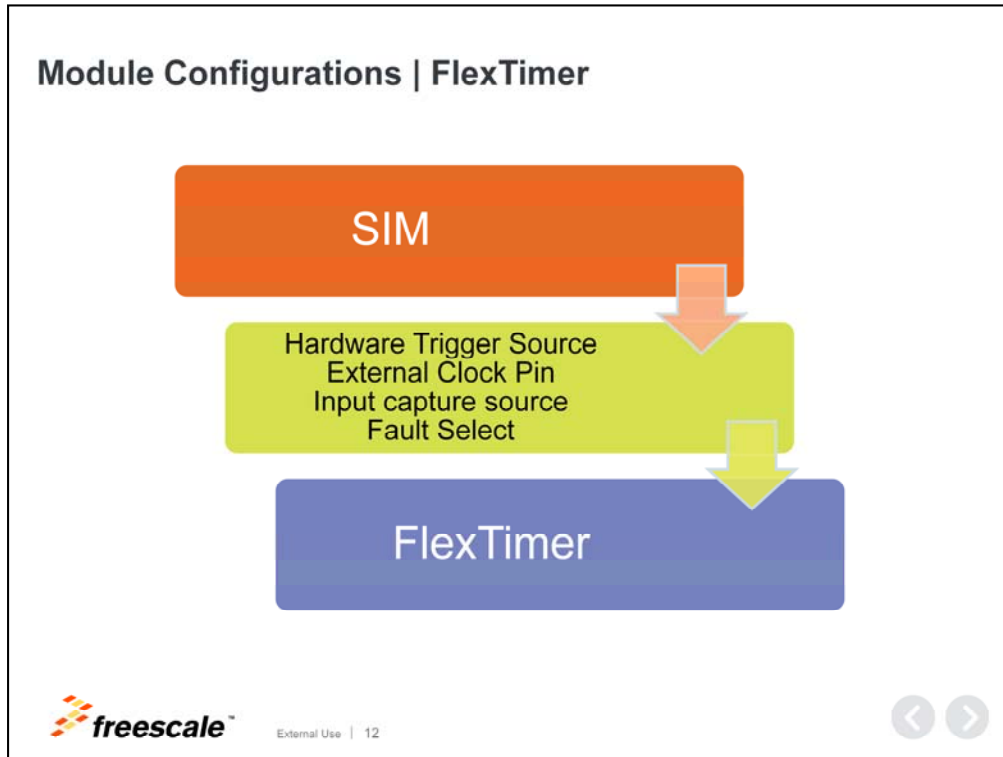
- Core clock
- Bus clock
- FlexBus clock
- And Flash clock



### Clock Gating Control

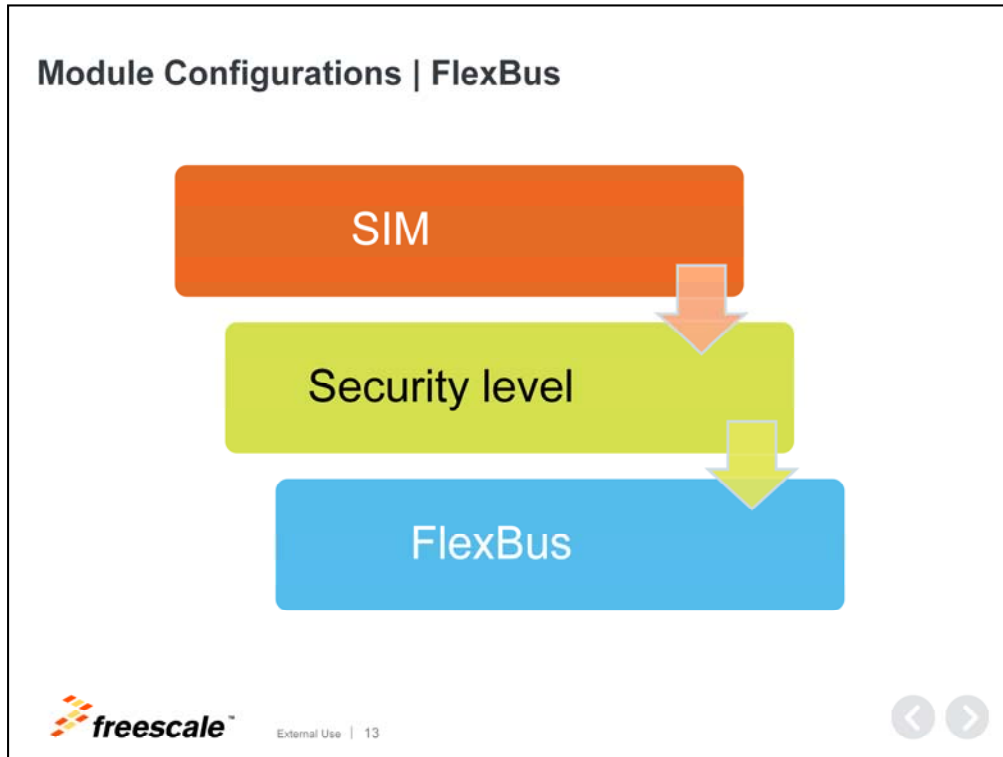
By default, most module clocks are disabled. The SIM allows you to enable or disable the clock gate to most modules on a per module basis.

To reduce power consumption, disable the clock for modules that are not used.



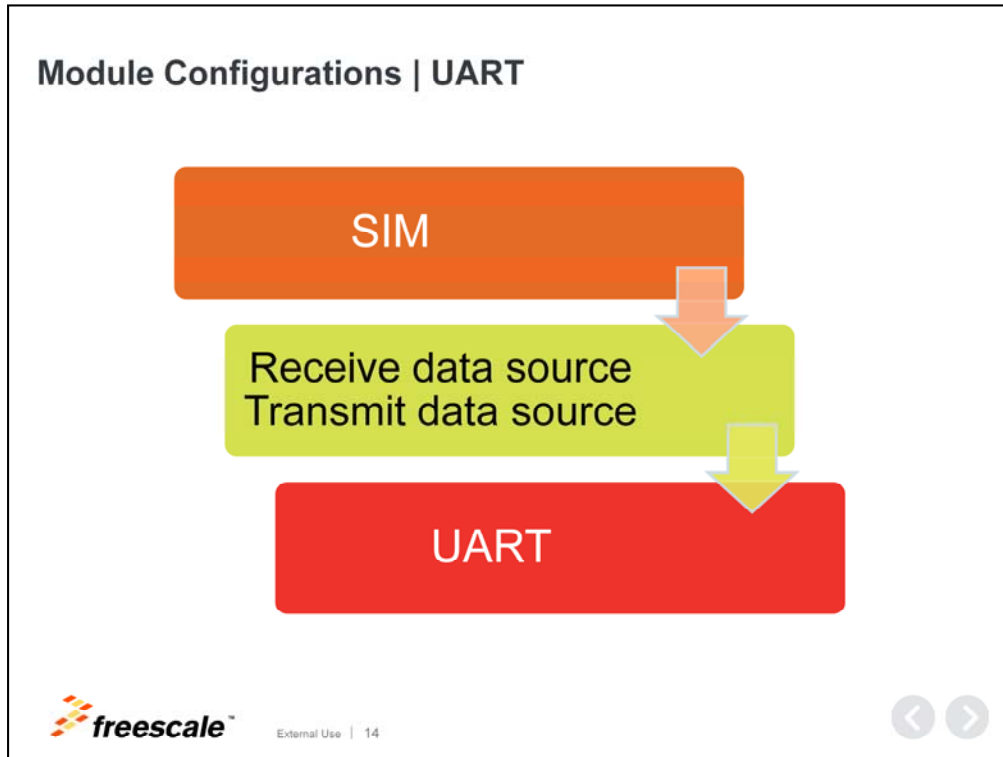
### Module Configurations | FlexTimer

The SIM allows you to configure several module settings. For example, **the Flextimer hardware trigger source, the external pin used to drive the clock or the input capture source and fault source.**



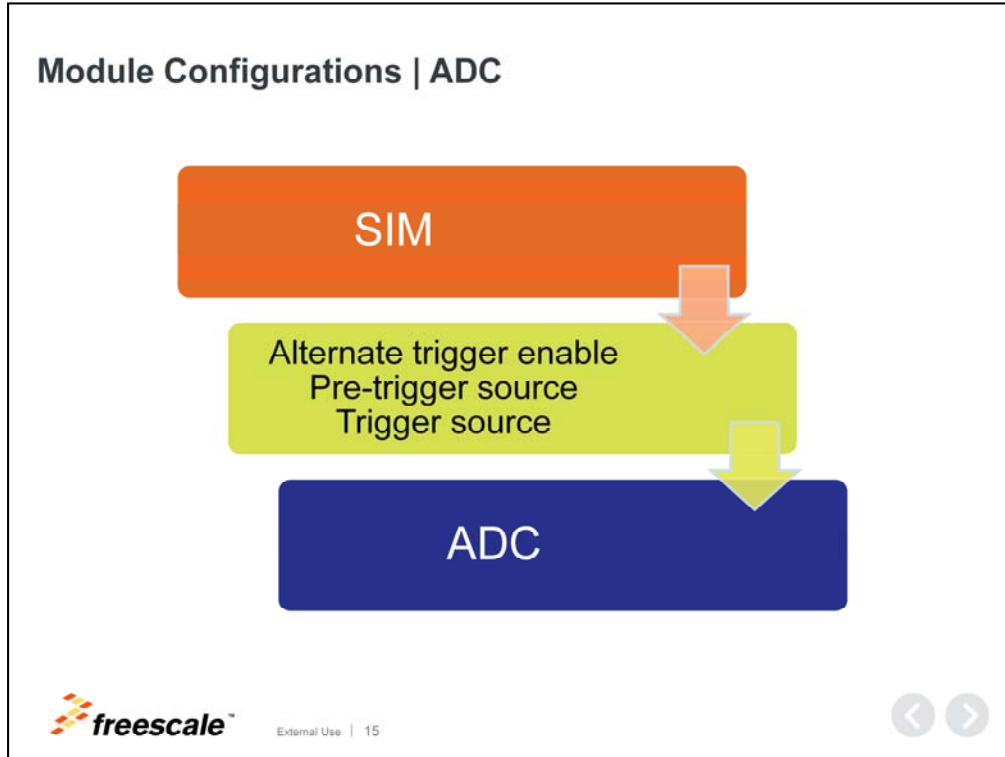
### Module Configurations | FlexBus

The SIM allows you to configure several module settings. For example, **the security level of the FlexBus.**



### Module Configurations | UART

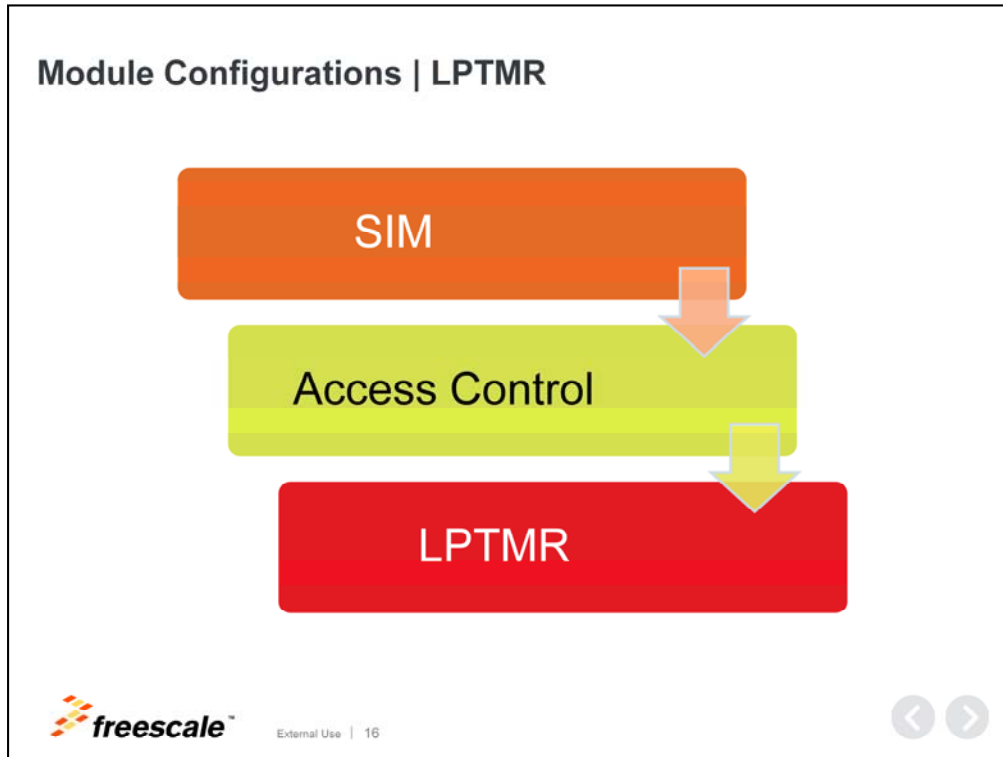
The SIM allows you to configure several module settings. For example, **selecting the UART receive and transmit sources.**



### Module Configurations | ADC

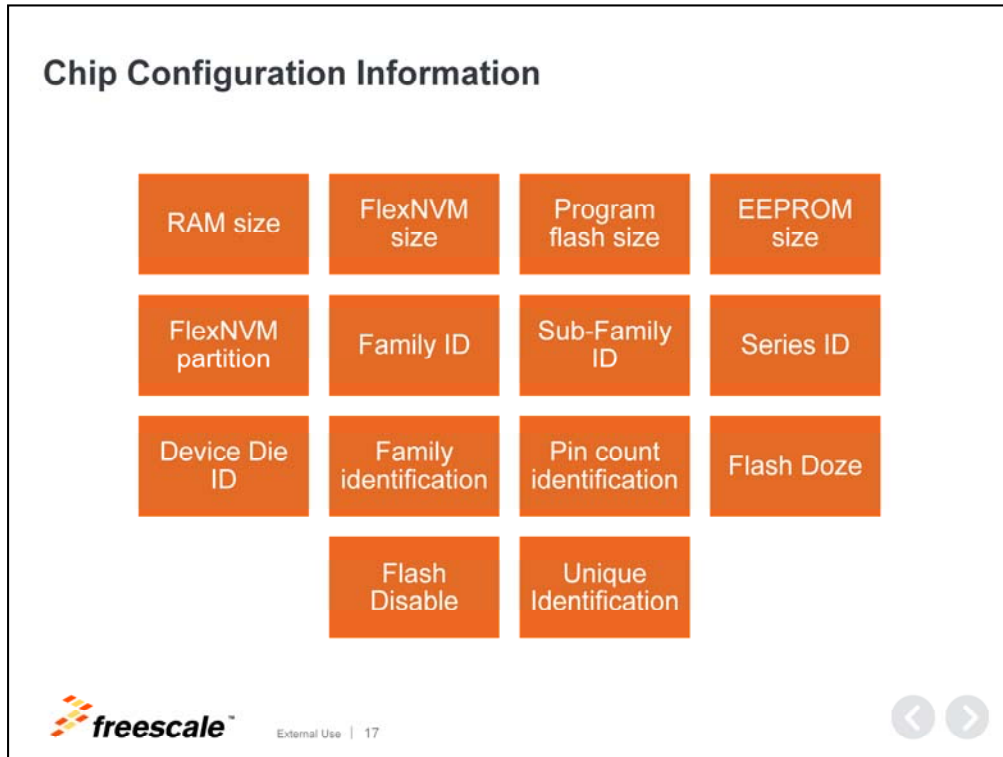
The SIM allows you to configure several module settings. For example, **selecting alternative conversion triggers, trigger and pre-trigger sources for the ADC module.**





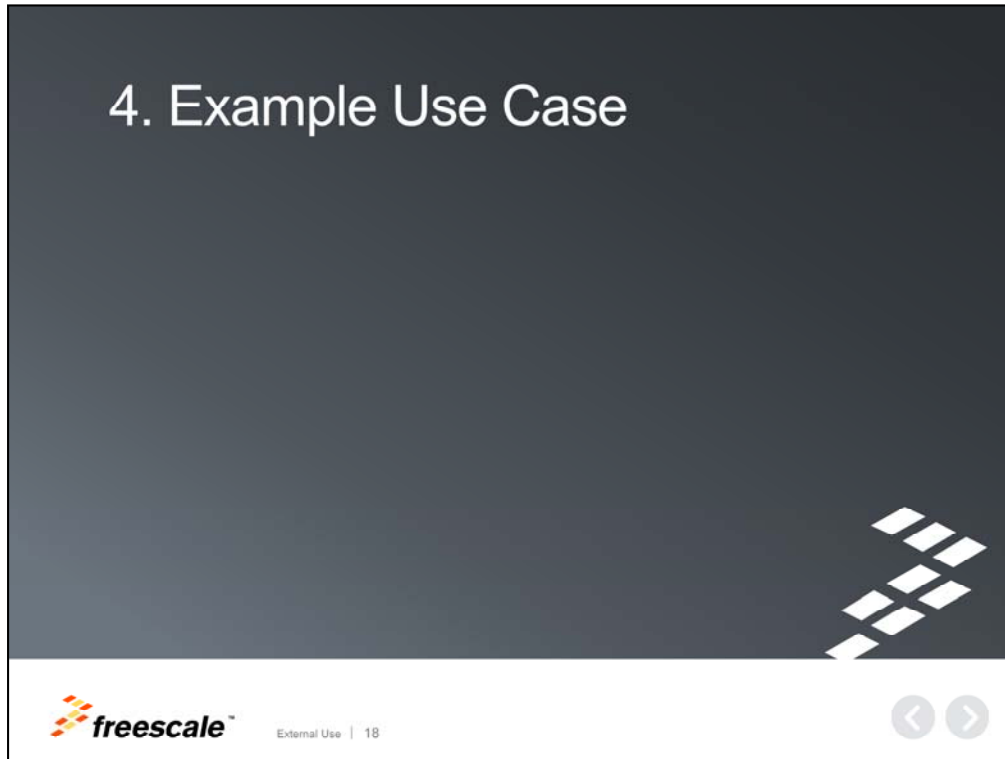
### Module Configurations | LPTMR

The SIM allows you to configure several module settings. For example, **controlling the software access to the low-power timer module (LPTMR)**.



## Chip Configuration Information

The SIM stores chip configuration information. This information can be useful if you have multiple versions of product with slightly different feature sets. For example, if you have one product with USB and another without USB, then at run time, your code could use the family ID value to determine which version of the product is running. This makes code maintenance and releases easier to track. However, this also makes the overall code size larger due to the inclusion of code for features that might not be supported.



Now, let's review an example use case.

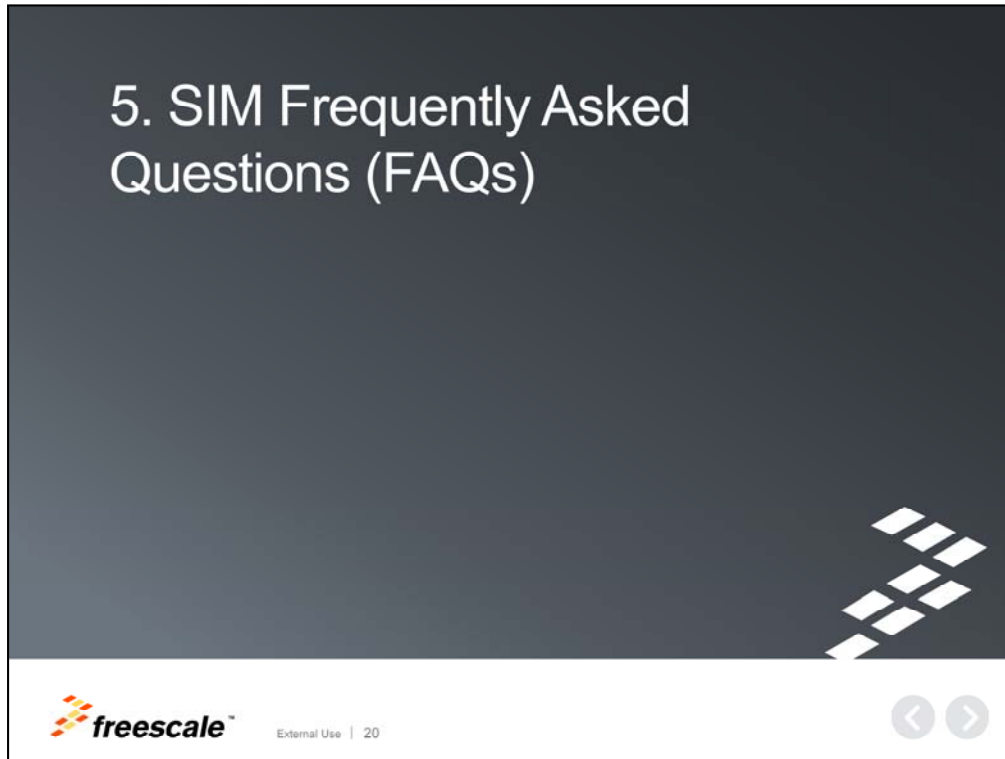
## Example Use Case

To configure the I/O pin muxing options, the port clocks must be enabled first.

```
/* enable clock for PORTs */  
CLOCK_SYS_EnablePortClock(PORTA_IDX);  
CLOCK_SYS_EnablePortClock(PORTB_IDX);  
CLOCK_SYS_EnablePortClock(PORTC_IDX);  
CLOCK_SYS_EnablePortClock(PORTD_IDX);  
CLOCK_SYS_EnablePortClock(PORTE_IDX);
```

## Example Use Case

In this example, we enable the MCU port clocks A, B, C, D and E by using the `CLOCK_SYS_EnablePortClock` function of Kinetis SDK. This allows the pin functions to be configured at a later time.



Finally, some frequently asked questions about the SIM module:

## SIM Frequently Asked Questions (FAQs)

**Q. Do I have to enable the clock to a module before I initialize it?**

A. Yes. The module clock must be enabled to access registers. Enabling the module clock should be one of the first steps in your module initialization, and in most cases it will be the first instruction. If you do not enable the clock to the module first, you will get a hard fault.

**Q. Do I need to enable the port module clocks to configure pin functions?**

A. Yes. The port module clocks have to be enabled in order to access port control registers (PCRs) to change pin function selections. You can disable the port clock once you have made PCR changes.

**Q. Do I need to update the SIM module registers if porting from one Kinetis device to another?**

A. Yes. This is necessary as the modules and interconnections can vary from one Kinetis MCU to another.



External Use | 21



## SIM Frequently Asked Questions (FAQs)

**Q: Do I have to enable the clock to a module before I initialize it?**

A: Yes, the module clock must be enabled to access registers. Enabling the module clock should be one of the first steps in your module initialization, and in most cases it will be the first instruction. If you don't enable the clock to the module first, you will get a hard fault.

**Q: Do I need to enable the port module clocks to configure pin functions?**

A: Yes, the port module clocks have to be enabled in order to access port control registers, or PCRs, to change pin function selections. You can disable the port clock once you've made PCR changes.

**Q: Do I need to update the SIM registers if porting from one Kinetis device to another?**

A: Yes. This is necessary as the modules and interconnections may vary from one Kinetis MCU to another.

## References

- **Website:** [Freescale.com/Kinetis](https://www.freescale.com/Kinetis)
- **Community:** [community.freescale.com/community/Kinetis](https://community.freescale.com/community/Kinetis)  
[community.freescale.com/message/326254](https://community.freescale.com/message/326254)

This concludes our presentation on the SIM module for Kinetis K series MCUs.

We invite you to visit us on the web at [Freescale.com/Kinetis](https://www.freescale.com/Kinetis) and don't forget to check out our Kinetis community pages for more information and discussions on this topic.



[www.Freescale.com](http://www.Freescale.com)

© 2015 Freescale Semiconductor, Inc. | *External Use*