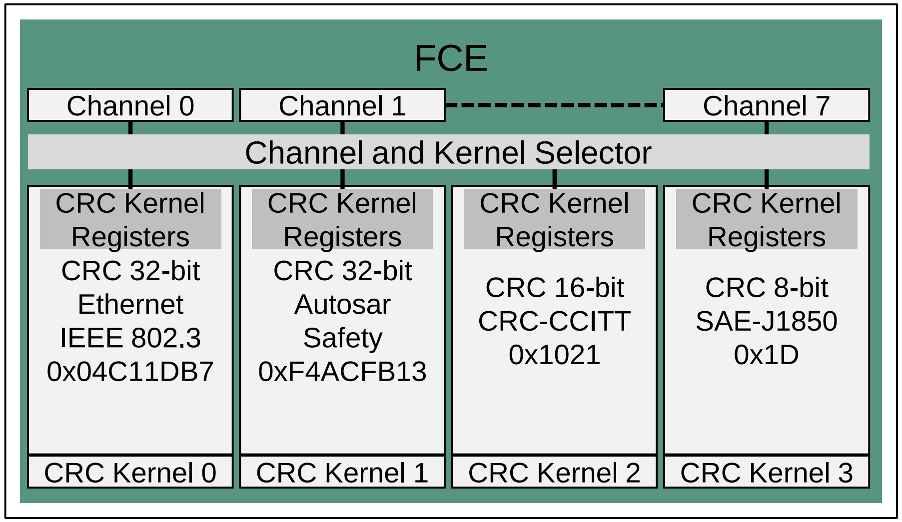
FCE

Flexible CRC Engine

AURIX™ TC3xx Microcontroller Training
V1.0 2020-09

FCE
Flexible CRC Engine



Highlights

- The Flexible CRC Engine FCE is used to compute cyclic redundancy checksums without CPU intervention
- Parallel CRC implementation calculates CRC checksum of a word within 1 SPB clock cycle
- 8 CRC channels that can be used with any of the kernels

Key Features

Multiple CRC polynomial kernels

Configurable CRC parameters

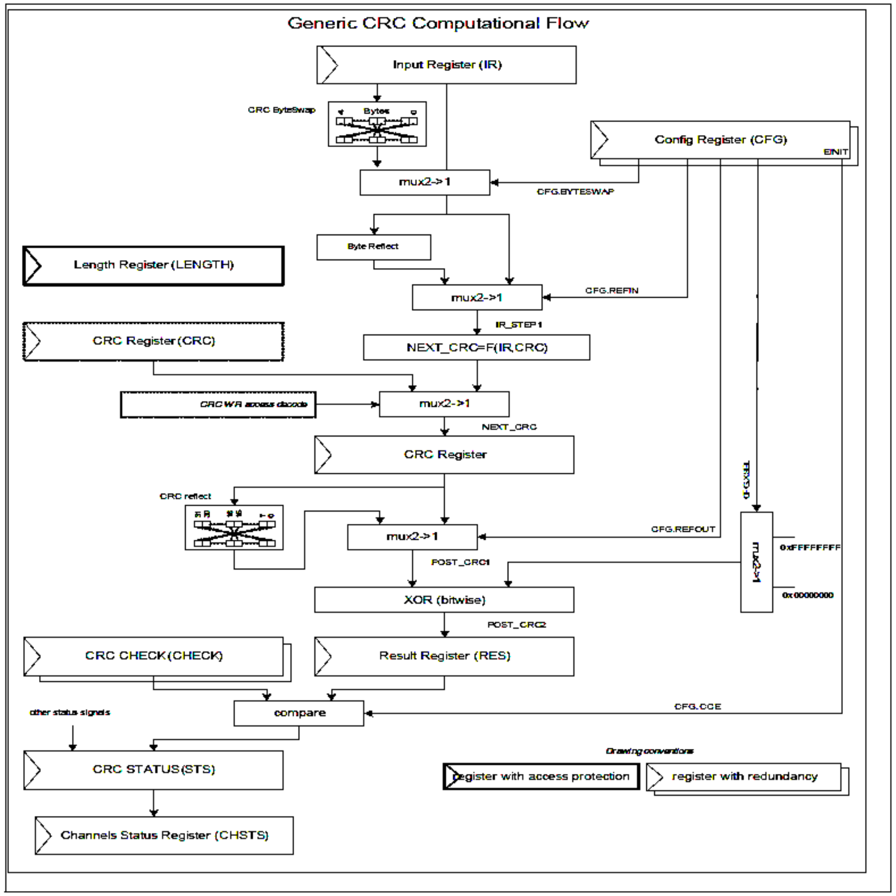
Automatic checksum checks

Customer Benefits

- Different CRC variants are supported: CRC32, CRC16 and CRC8
- CRC algorithms can be adapted to the application needs
- Automated comparison of expected vs. calculated checksum

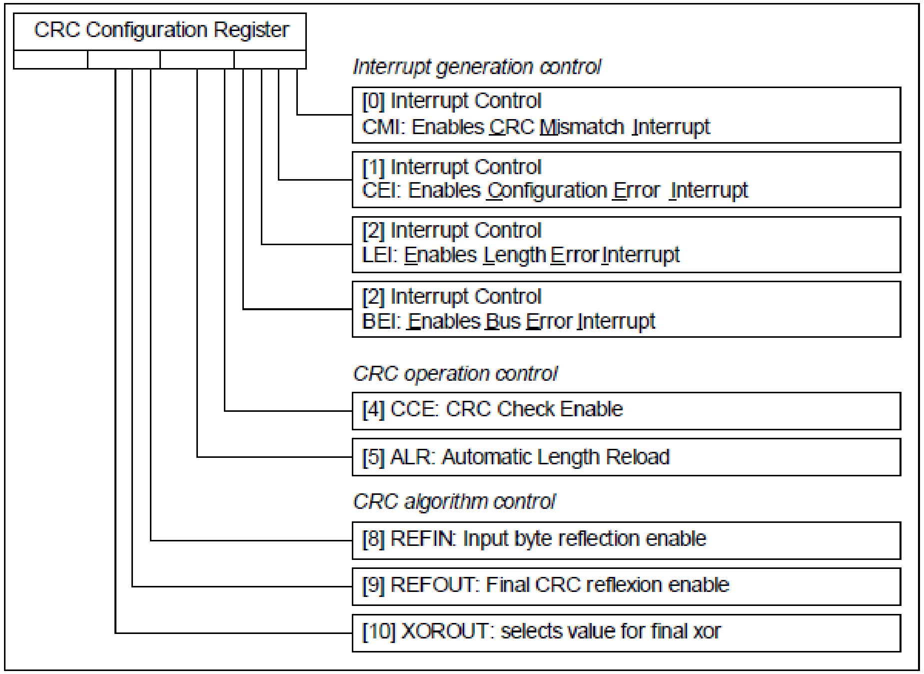
FCE
Multiple CRC polynomial kernels

- The generic architecture of an FCE CRC Kernel is shown on the right
- 4 such kernels are supported:
  - Kernel 0 : IEEE 802.3 CRC32 Ethernet polynomial: 0x04C11DB71
  - Kernel 1 : Autosar safety CRC32P4 polynomial: 0xF4ACFB13
  - Kernel 2 : 16-bit CRC-CCITT polynomial: 0x1021
  - Kernel 3: SAE J1850 CRC8 polynomial: 0x1D
- The usage of the kernel:
  - The input values need to written to the IR register
  - After 2 clock cycles, the calculated CRC result is available in the RES register



FCE
Configurable CRC parameters

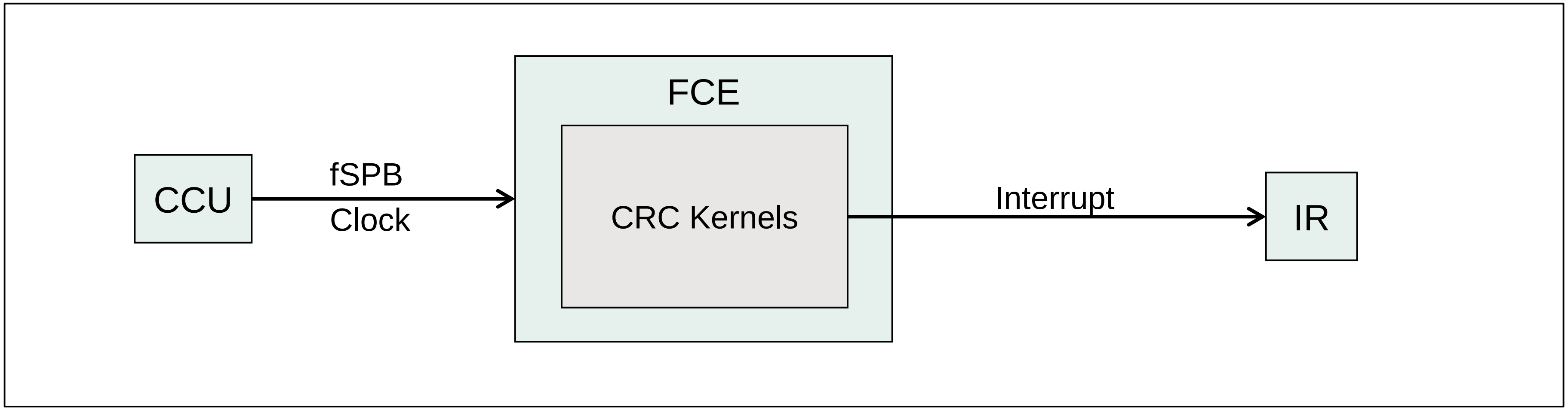
- The supported configurations for each kernel are shown on the right
- The length of the message can be configured
- For the CRC computation, the following configurations are important:
  - Input byte reflection
  - Output bit reflection
  - Output XOR (inversion)



FCE
Automatic checksum checks

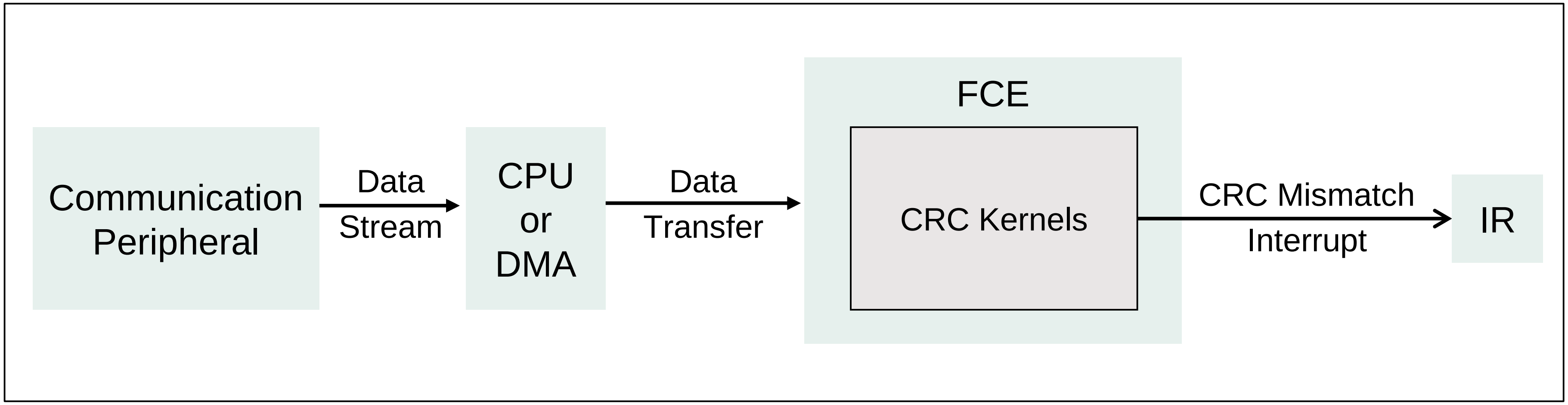
- The FCE supports an automatic checksum checks at the end of a message
- This means the FCE can be programmed to generate an interrupt, in case the CRC result does not match an expected CRC value

FCE
System integration



› The FCE
  - gets its clock from the System Peripheral Bus clock (f_{SPB})
  - provides one interrupt line to the interrupt router (IR) indicating:
    - CRC mismatch
    - configuration error
    - length error
    - bus error

Application example
CRC computation



Overview

FCE can be used to accelerate CRC computation.
For example, a data stream from a communication peripheral is fed to FCE via DMA or CPU.

Advantages

- Usage of DMA offloads the CPU
- Automatic CRC check at the end of computation
- CRC32 results from FCE and TriCore™ instruction are identical

Edition 2020-09
Published by
Infineon Technologies AG
81726 Munich, Germany

Do you have a question about this document?
Email: erratum@infineon.com

Document reference
AURIX_Training_2_Flexible_CRC_Engine