AN13198

OTFAD performance i.MXRT1170

Rev. 0 — 19 May 2021 Application Note

Contents

1 Introduction....1
2 OTFAD module....2
3 Measurement....4
4 Measurement results....6
5 Conclusion....6
6 References....7
7 Revision history....7

1 Introduction

1.1 Purpose

The purpose of this application note is to provide a comparison of external flash read speeds when unencrypted and when the OTFAD module is used for decryption.

1.2 Intended audience

This document is intended for those who need an overall idea about the OTFAD decryption performance. It does not explain how to setup an encrypted XIP OTFAD image. It is assumed that the reader is familiar with the basics of encrypted XIP provided by the OTFAD module on i.MX RT1170. It is assumed that the reader is already familiar with the SPT Secure Provisioning Tool.

1.3 Scope

This document is a practical example that provides the measurement results of an unencrypted XIP compared to the encrypted XIP read performance.

1.4 Acronyms and abbreviations

The terms and acronyms used in this document are:

- AES – Advanced Encryption Standard.
- AHB – internal bus connected to a FlexSPI module.
- AXI – internal bus connected to an L1 cache. The AXI and AHB are connected with the AHB/AXI bus convertor.
- FlexSPI – NXP proprietary module to access Single/Dual/Quad/Octal/Hyperbus and similar serial-bus-based devices.
- XIP – Execute-In-Place. It refers to a software image that is executed directly from its non-volatile memory.
- Unencrypted XIP – refers to a software image that is executed directly from the non-volatile memory and there is no need to use any decryption.
- Normal XIP – refers to the unencrypted XIP.
- Encrypted XIP – refers to an encrypted software image that is executed directly from its non-volatile memory and it must be decrypted by the OTFAD module.
- OTFAD – On-The-Fly AES Decryption module.
- SPT – Secure Provisioning Tool, which provides an encrypted XIP image and uploads the image to the target board.
- MCUX IDE – MCUXpresso IDE. It is an easy-to-use Integrated Development Environment (IDE) for creating, building, debugging, and optimizing of application code.
- ITCM – internal memory accessed by its own I-TCM bus interface (single-cycle memory recommended for instruction fetches - code execution, interrupt vector table).

- DTCM – internal memory accessed by its own D0-TCM/D1-TCM interface (single-cycle memory recommended for data access - stack, important static variables).
- I-CACHE
  - Read cache hit: represents a single-cycle memory for instruction fetches.
  - Read cache miss: generates a 32-B burst transfer on an AXI bus.
- D-CACHE
  - Read cache hit: represent a single-cycle memory for data accesses.
  - Read cache miss: generates a 32-B burst transfer on an AXI bus.
- OCRAM – internal memory accessed by an AXI bus via an interconnect bus fabric NIC. It is expected to be cached to achieve sufficient performance. Multiple wait states are generated when it is accessed via the AXI. The data is accessed by multiple masters, such as CM7 and DMA. Avoid placing a stack here if possible.

2 OTFAD module

The On-The-Fly AES Decryption (OTFAD) module provides an advanced hardware implementation that minimizes any incremental cycles of latency introduced by the decryption in the overall external memory-access time. It implements a block cipher mode of operation supporting the counter mode (CTR). The CTR mode provides a confidentiality mode that features the application of the forward cipher to a set of input blocks (called counters) to produce a sequence of output blocks that are exclusive-ORed with the plaintext to produce the ciphertext and vice versa.

The OTFAD engine includes complete hardware support for a standard AES key unwrap mechanism to decrypt a key BLOB data instruction containing the parameters needed for up to 4 unique AES contexts. Each context has a unique 128-bit key, a 64-bit counter, and a 64-bit memory region descriptor.

2.1 Basic OTFAD module features

- AES-128 counter mode on-the-fly decryption.
  - 128-bit key and 128-bit data block sizes.
  - The 128-bit counter includes 64 bits of the initialization vector plus the 32-bit system address.
- It adds zero cycles of incremental latency for decryption when used with the FlexSPI.
  - It receives 64-bit encrypted data from the FlexSPI, calculates the decrypted data which is sent to the AHB RAM buffer and bypassed back to the system AHB read data bus.
- Hardware support for 4 independent decryption segments (called memory “contexts”).
  - Each context has a unique 128-bit key, a 64-bit counter, and a 64-bit memory region descriptor.
- It functionally acts as a slave submodule to the FlexSPI.
  - It is logically connected between the FlexSPI and its AHB RAM buffer.
  - It shares the system AHB and IPS (slave peripheral) bus connections.
  - The programming model is mapped into the upper 1 KByte of the FlexSPI's IPS address space.
  - Private 64-bit data buses for encrypted (ciphertext) and decrypted (plaintext) data.
- Hardware microarchitecture.
  - Heavily pipelined AES engine optimized for encryption, performing 3 rounds per cycle.
  - 64-bit AHB connections for easy integration to the system bus fabric and FlexSPI.
  - Data storage for two 128-bit encrypted counters and three 64-bit decrypted data buffers.
  - Optimized for {32,64}-bit WRAP4 bursts (CPU cache miss fetch size and typical DMA fetch size).

2.2 OTFAD block diagram

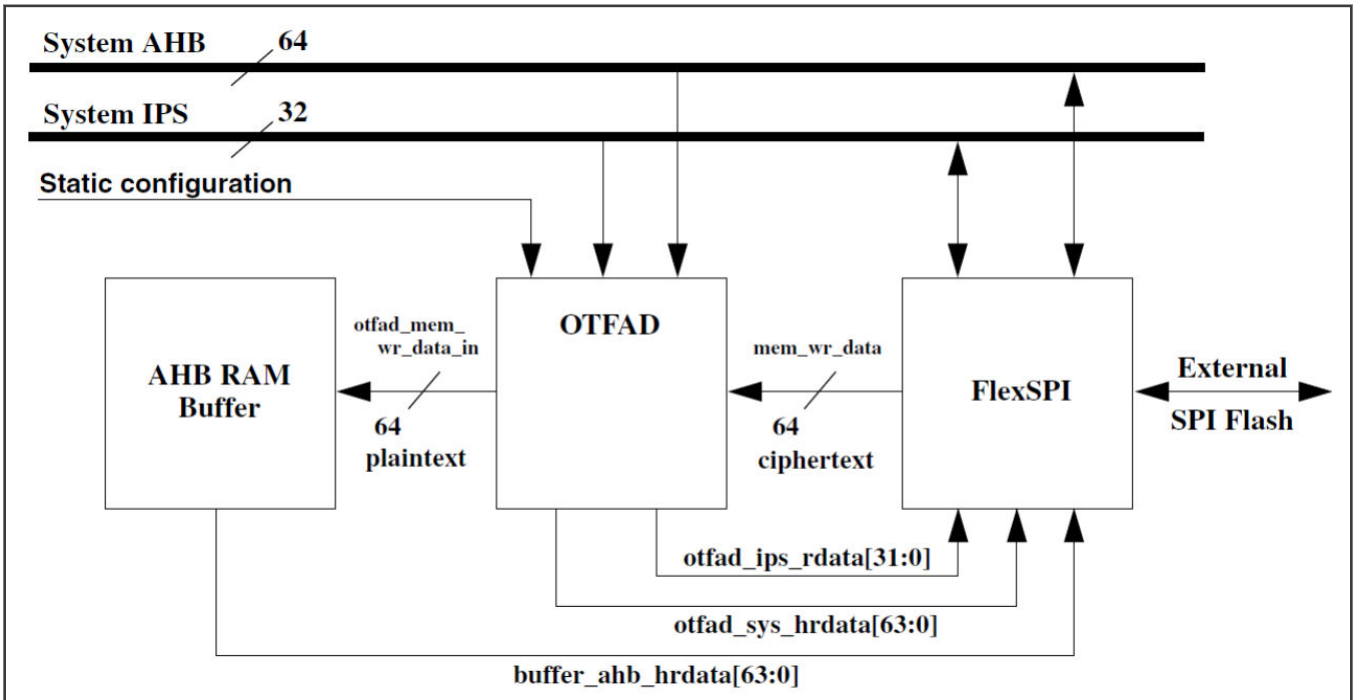

Figure 1. OTFAD and FlexSPI connection topology

2.3 OTFAD bus timing

This is an example of a 64-bit WRAP4 read request with an original address of 0x00.

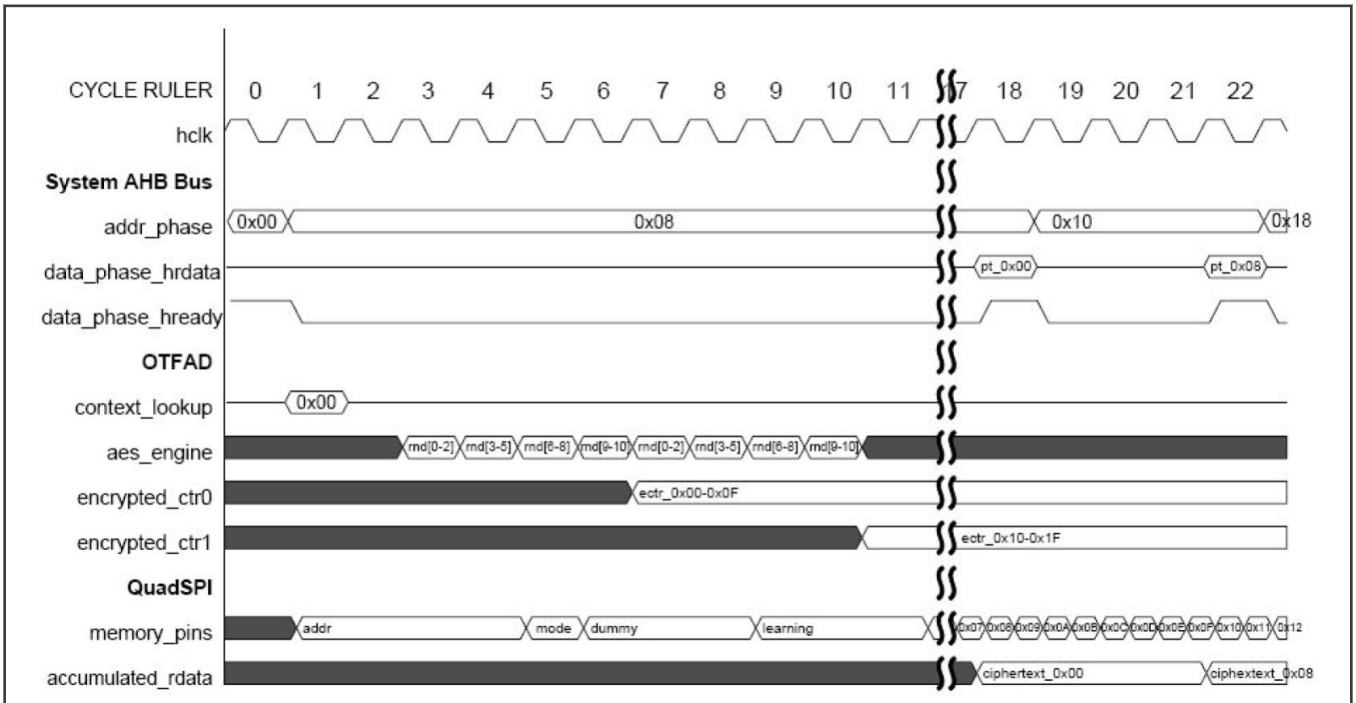

Figure 2. OTFAD 64-bit WRAP4 burst timing diagram

All the "encrypted_ctr0" and "encrypted_ctr1" values are being prepared by the AES engine during the time when the command, address, and modes bits are transferred on the FlexSPI bus pins and they are ready for an XOR operation when the data are being read from the QSPI memory. This conforms to the AES-CTR mode implementation, which is accommodated by the OTFAD module.

The resulting behavior of the combined FlexSPI and OTFAD provides the best system performance.

3 Measurement

3.1 Hardware requirements

The measurement was done using the MIMXRT1170-EVK board. The FlexSPI1 module port A was used, because it is routed to the QSPI memory by default.

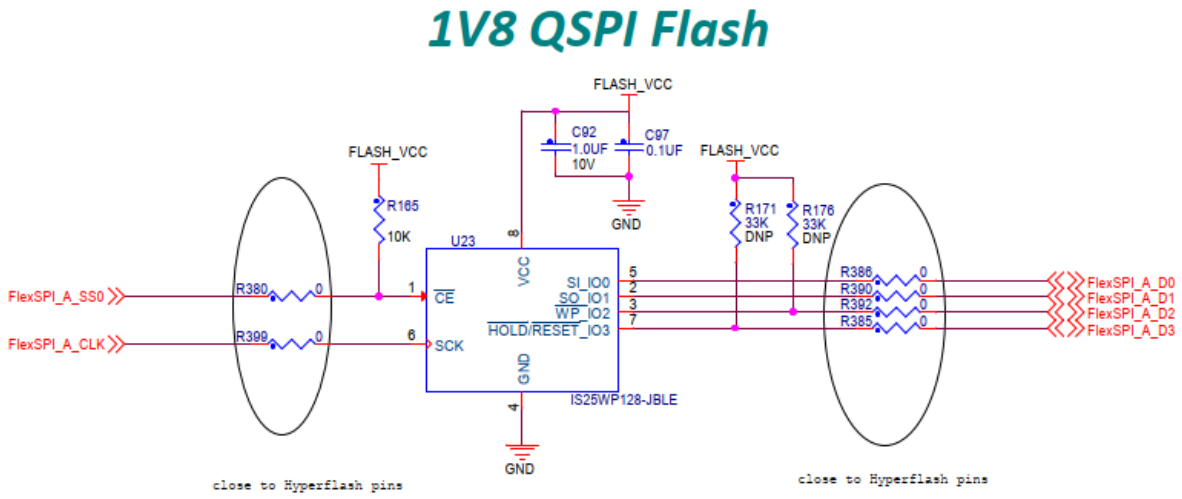

Figure 3. QSPI flash connection to FlexSPI1 port A

3.2 Software tools

The software used is as follows:

- MCUXpresso IDE v11.3.0 [Build 5222]
- "evkmimxrt1170_flexspi_OTFAD_performance_cm7" test application code – software provided along with this document
- Security Provisioning Tool v3.1

3.3 Measurement approach

The aim of the measurement is to compare the data-reading speed from the flash memory with and without the OTFAD module decryption.

The measurement itself is done by the MCUX IDE application "vkmimxrt1170_flexspi_OTFAD_performance_cm7". The application code allocates two read buffers in the read-only flash memory address space and measures the time of reading from these read buffers using a cycle counter from the CM7 core.

To get both results, the image of the measurement application is uploaded as a normal XIP in the first case and as an encrypted XIP (OTFAD mode) in the second case. In both cases, the same image is uploaded. Therefore, the same setting of the FlexSPI module is applied for the normal XIP and for the encrypted XIP. Only the OTFAD setting provided to the boot ROM code is different.

To create an encrypted XIP image and to upload it into the target board, the Secure Provisioning Tool (SPT) v3.1 must be used.

3.4 Measuring application

The application code is based on the "evkmimxrt1170_flexspi_nor_polling_transfer_cm7" SDK example code.

There are two 16-KB read-only buffers located in the flash filled with the incremental value from 0 – 16383.

```
const uint32_t text_read_buffer1[] __attribute__((aligned(1024))) = { T_FILL4096(0) };
const uint32_t text_read_buffer2[] __attribute__((aligned(1024))) = { T_FILL4096(0) };
```

3.4.1 FlexSPI module settings

The tests were made with the 1V8 QSPI flash memory IS25WP128-JBLE. The FlexSPI setting used for the SDK flash driver is as follows:

- Use the SDR/READ_FAST_QUAD for the default AHB access.
- Enable the Prefetch buffer with a size of 4 KB for the core.
- Enable the FlexSPI root clock to 99 MHz.

3.4.2 Measuring functions

- The "flexspi_nor_readData_8b_itcm" function measures the number of core cycles to read from "text_read_buffer1" using an 8-bit access:

```
__ASM volatile ("LDRB.W r3,[r0],#1\n");
__ASM volatile ("LDRB.W r3,[r0],#1\n");
```

- The "flexspi_nor_readData_pingpong_8b_itcm" function measures the number of core cycles to read from "text_read_buffer1 and text_read_buffer2" alternately using an 8-bit access:

```
__ASM volatile ("LDRB.W r3,[r0],#1\n");
__ASM volatile ("LDRB.W r3,[r1],#1\n");
```

- The "flexspi_nor_readData_16b_itcm" measures the number of core cycles to read from "text_read_buffer1" using a 16-bit access:

```
__ASM volatile ("LDRH.W r3,[r0],#2\n");
__ASM volatile ("LDRH.W r3,[r0],#2\n");
```

- The "flexspi_nor_readData_pingpong_16b_itcm" function measures the number of core cycles to read from "text_read_buffer1" and "text_read_buffer2" alternately using a 16-bit access:

```
__ASM volatile ("LDRH.W r3,[r0],#2\n");
__ASM volatile ("LDRH.W r3,[r1],#2\n");
```

- The "flexspi_nor_readData_32b_itcm" function measures the number of core cycles to read from "text_read_buffer1" using a 32-bit access:

```
__ASM volatile ("LDR.W r3,[r0],#4\n");
__ASM volatile ("LDR.W r3,[r0],#4\n");
```

- The "flexspi_nor_readData_pingpong_32b_itcm" function measures the number of core cycles to read from "text_read_buffer1" and "text_read_buffer2" alternately using a 32-bit access:

```
__ASM volatile ("LDR.W r3,[r0],#4\n");
__ASM volatile ("LDR.W r3,[r1],#4\n");
```

- The "flexspi_nor_readData_burst_itcm" function measures the number of core cycles to read from "text_read_buffer1" using a burst access:

```
__ASM volatile ("LDM r1,{r4-r11}\n");
```

All these functions are located and executed in the ITCM to not influence the read speed from "text_read_buffer".

3.4.3 L1 D-CACHE setting

The measurement functions are executed in the following three ways in regards to the L1 D-CACHE setting:

1. D-CACHE is disabled. The cache is disabled before the measurement.
2. D-CACHE is invalidated. The D-CACHE is enabled and invalidated before each measurement.
3. D-CACHE is filled. The D-CACHE is enabled and filled with measurement values when the measurement is made for the first time. The next measurement is made for the second time without invalidating the D-CACHE and it is expected to provide a 100 % cache hit. The value of the second measurement is returned.

3.4.4 Buffers size setting

The whole measurement is made for the "text_read_buffer1" and "text_read_buffer2" size set to 4 KB and 16 KB. The application must be built for each buffer size setting.

4 Measurement results

Cortex M7 - CORE @[illegible], QSPI @[illegible] SDR								
			4K read buffer			16K read buffer		
			D-CACHE			D-CACHE		
			Disabled	Invalidated	Filled	Disabled	Invalidated	Filled
Standard FlexSPI access	Linear READ	8-bit	18.29	47.64	995.51	18.38	47.73	995.88
		16-bit	36.72	47.71	1990.06	36.74	47.74	1991.51
		32-bit	47.68	47.71	3976.23	47.72	47.75	3982.06
		BURST 8	47.71	47.65	5277.64	47.75	47.73	5303.37
	PingPong READ	8-bit	1.46	25.9	1061.85	1.46	27.03	1062.19
		16-bit	2.91	25.71	2274.03	2.93	26.98	2275.62
		32-bit	5.83	25.82	4710.87	5.89	27.01	4588.99
		BURST 8	-	-	-	-	-	-
OTFAD FlexSPI access	Linear READ	8-bit	19.65	46.93	995.51	19.76	47.01	995.88
		16-bit	39.14	47.01	1990.06	39.16	47.05	1991.51
		32-bit	46.91	47.01	3976.23	46.93	47.05	3982.06
		BURST 8	47.01	46.95	5227.64	47.05	47.03	5303.37
	PingPong READ	8-bit	1.46	27.71	1061.85	1.46	27.73	1062.19
		16-bit	2.93	27.63	2274.03	2.93	27.71	2275.62
		32-bit	5.96	27.63	4710.87	5.96	27.71	4588.99
		BURST 8	-	-	-	-	-	-

Figure 4. Measured transfer speed in Mbytes/s

5 Conclusion

The results are consistent with the hardware specification of the FlexSPI connection to the internal 64-bit AHB bus. The best results are acquired using a 32-byte burst access. When the D-CACHE is invalidated, it means that it does not contain any cached data, so an 8-bit or 16-bit access is accelerated significantly. This is because an incoming 8-bit request is transferred to the AXI/AHB bus as a 32-byte request. The cache is filled with a 32-byte response value and the next 31 8-bit read requests are hits in the D_CACHE.

The final result is that the OTFAD module does not provide any significant performance decrease, which can be measured by this application. This is achieved by the OTFAD module design and implementation of the AES128-CTR decryption mode.

6 References

1. MXUXpresso IDE - https://www.nxp.com/design/software/development-software/mcuxpresso-software-and-tools-/mcuxpresso-integrated-development-environment-ide:MCUXpresso-IDE
2. Secure Provisioning Tool - https://www.nxp.com/design/software/development-software/mcuxpresso-software-and-tools-/mcuxpresso-secure-provisioning-tool:MCUXPRESSO-SECURE-PROVISIONING?tid=vanMCUXPRESSO-SECURE-PROVISIONING
3. *i.MX RT1170 Processor Reference Manual* (document IMXRT1170RM)
4. *Security Reference Manual for the i.MX RT1170 Processor* (document IMXRT1170SRM)
5. *Using the i.MXRT L1 Cache* (document AN12042)
6. AN memory benchmark performance by RP *****please provide the document ID)*****

7 Revision history

Table 1. Revision history

Revision number	Date	Substantive changes
0	19 May 2021	Initial release

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

For more information, please visit: http://www.nxp.com

For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 19 May 2021

Document identifier: AN13198

arm