AN13097

Using External Hardware Trigger to Initiate ADC Conversion on i.MX RT1170

Rev. 0 — 02/2021 Application Note

Contents

1 Introduction

NXP has implemented a new ADC module on i.MX RT1170, which was not present on it's the previous crossover family members, such as iMXRT1050/RT1060. The main reason to do so is to increase the sample rate. This ADC module is not an update of the ADC module used on previous RT devices, but it is a completely different module. To ensure compatibility with the previous devices, there are upgrades in modules which have basically the same functionality as on the previous devices. This upgrade can be potentially invisible, but it may require an explanation.

This document deals with the ADC hardware triggering capability implemented on iMXRT1170. There is only one hardware trigger source for the ADC on iMXRT1170. It is the ADC external trigger control (ADC_ETC). Its functionality is basically the same as on the previous RT devices. However, NXP completely changed the ADC module on iMXRT1170 to achieve a higher sample rate. Because of this NXP upgraded the ADC module with a logic which allows to interact with the ADC_ETC module to achieve the same functionality as on the previous iMXRT 4-digit devices.

2 ADC hardware trigger interconnection

To trigger the ADC on iMXRT1170 by a hardware signal, it is required to properly configure the following three peripheral modules:

- Inter-peripheral crossbar switch (XBAR)
- ADC external trigger control module (ADC_ETC)
- ADC (LPADC)

Each of those modules must be properly configured to allow the ADC module to start the conversion. The initial trigger must always come from an input signal of the XBAR. The rising edge timing must follow the minimum trigger time specification (one ADC bus clock). For example, it means that the XBAR1_IN01 signal, which is permanently connected to LOGIC HIGH, cannot be used as a trigger source. After the initial trigger coming from the XBAR, it is possible to generate up to 7 additional triggers in a chain without XBAR module utilization. They are generated by the ADC_ETC module itself. In such case, each next (chain) trigger is generated based on the following:

- The delay interval configurable by the ADC_ETC module. This time interval must follow the rule of the minimum time interval value, which is defined by the total conversion time. If the time interval is lower than the total conversion time, the trigger is generated during an active conversion and ignored. In such case, it is possible to generate an error interrupt event.
- The conversion-complete event generated by the ADC module. In such case, the back-to-back mode within the ADC_ETC module is configured. This allows to perform sequential ADC conversions without a restriction in the delay interval time configuration. The delay time is generated automatically, based on the waiting for the conversion-complete signal generated by the ADC module at the end of the conversion.

After the last sequential (chained) trigger generated by ADC_ETC, it is again required to generate the initial trigger via XBAR. It is also possible to generate two completely synchronous trigger chains for two available ADC modules. This approach allows to generate 8 sequential conversions by one ADC1 module and additional 8 sequential conversions by another ADC module in parallel, (16 conversion in total) initiated by only 1 trigger source.

2.1 Inter-peripheral crossbar switch (XBAR)

This module allows to configure more than 150 peripheral signals to be the source of the ADC trigger. It includes 8 output signals dedicated for TRIG0 – TRIG7 of the ADC external trigger control module (ADC_ETC). These 8 output signals are split into two groups of 4 input signals for ADC_ETC. This means the following:

- ADC_ETC_XBAR0
  - TRIG0
  - TRIG1
  - TRIG2
  - TRIG3
- ADC_ETC_XBAR1
  - TRIG0
  - TRIG1
  - TRIG2
  - TRIG3

All these signals are utilized by the ADC_ETC module to generate an adequate hardware trigger for the ADC modules.

NOTE

The ADC_ETC dedicated trigger signals are only available in the XBAR1 module (see Figure 1).

The XBAR modules also utilize 8 input signals, which are output from the ADC_ETC module. These signals represent the conversion-complete signal dedicated by corresponding ADC_ETC hardware trigger channels TRIG0 – TRIG7. These 8 input signals are split into two groups of 4 output signals from ADC_ETC. This means the following:

- ADC_ETC0
  - COCO0
  - COCO1
  - COCO2
  - COCO3
- ADC_ETC1
  - COCO0
  - COCO1
  - COCO2
  - COCO3

These signals can be utilized as trigger events for other peripherals. However, they can still be utilized in a loopback to the ADC_ETC module, if the application use case requires it.

NOTE

Consider some latency on the XBAR between the input and output signals. The latency is not fixed, because XBAR utilizes combinational logic instead of clock-driven flip-flops. However, on RT1170, it is a couple of ns (up to 10 ns). This time is estimated based on simulations.

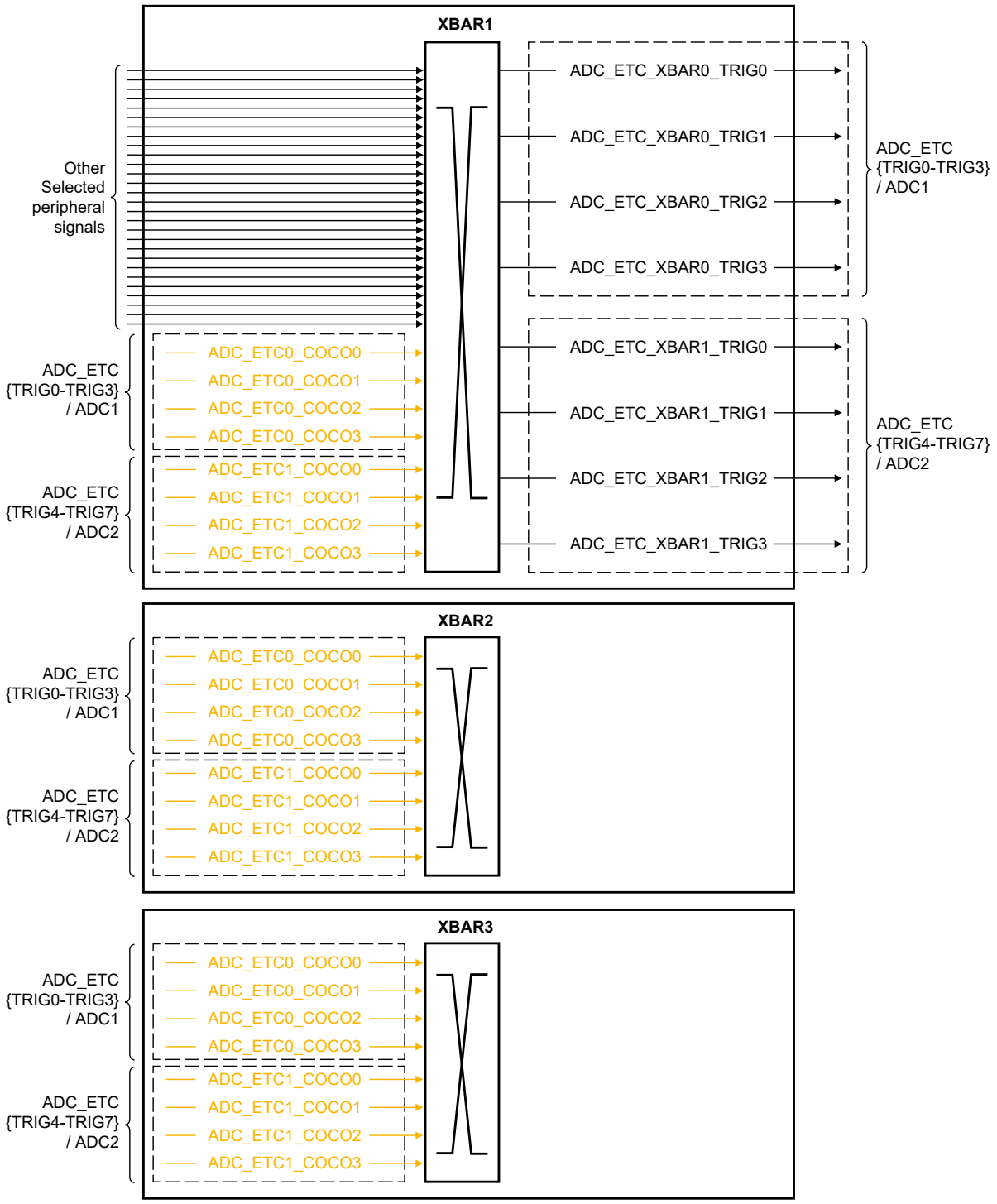

Figure 1. XBAR signals dedicated to ADC_ETC

2.2 ADC external trigger control (ADC_ETC)

The ADC_ETC module represents the key for ADC hardware triggering. This module was first implemented on RT1050 devices. Its primary role was dedicated to trigger the ADC, which can be found on NXP's legacy MCU devices, such as Kinetis V and DSCs. These devices have complex support of the ADC triggering for power-conversion and motor-control applications. It is not

limited only to those applications. The ADC_ETC module allows flexible and complex triggering of the ADC. Because the ADC module implemented on RT1050 has the sample rate limited to 1 MSps and the higher sample rate was needed, NXP implemented a completely new ADC module on RT1170. The ADC on the RT1170 supports sample rates of up to 4 MSps. However, this implementation required a wrapper on the ADC to follow all internal signals required by the ADC_ETC module. That is why the ADC_ETC module on the RT1170 has a different meaning in some configurations. However, the primary functionality of the module remains unchanged. It utilizes signals from the XBAR to generate the initial ADC trigger and it is possible to generate additional chained trigger signals (up to 8 trigger signals in total for one chain). It also allows to generate two synchronous trigger signals for both ADC modules available on the RT1170.

It utilizes one of 4 dedicated XBAR output signals within one of two groups to initiate trigger generation for one of the two ADC modules:

- Initial trigger for ADC1:
  - TRIG0 channel (primary source coming from ADC_ETC_XBAR0_TRIG0)
  - TRIG1 channel (primary source coming from ADC_ETC_XBAR0_TRIG1)
  - TRIG2 channel (primary source coming from ADC_ETC_XBAR0_TRIG2)
  - TRIG3 channel (primary source coming from ADC_ETC_XBAR0_TRIG3)
- Initial trigger for ADC2:
  - TRIG4 channel (primary source coming from ADC_ETC_XBAR1_TRIG0)
  - TRIG5 channel (primary source coming from ADC_ETC_XBAR1_TRIG1)
  - TRIG6 channel (primary source coming from ADC_ETC_XBAR1_TRIG2)
  - TRIG7 channel (primary source coming from ADC_ETC_XBAR1_TRIG3)

One trigger signal coming from the XBAR module can initiate up to 8 consecutive triggers signals generated by one trigger channel within the ADC_ETC module. The first trigger signal is synchronous with the XBAR signal generation and the next (up to 7) trigger signals can be generated by the chains (CHAIN1-CHAIN7) within one trigger channel (TRIGx) of the ADC_ETC module.

For example, ADC_ETC should be used to generate 8 sequential triggers to the ADC1, initiated by the trigger signal coming from the ADC_ETC_XBAR0_TRIG0. For an illustration, see Figure 2.

- The initial trigger signal is generated by the XBAR module output ADC_ETC_XBAR0_TRIG0. This trigger signal passes through the ADC_ETC arbiter and the synchronization logic (see Figure 4) to the ADC1. The ADC1 module initiates the first conversion, based on its trigger defined by the HWTS0 field (ADC trigger selection) and its command defined by the CSEL0 (ADC command selection) within the ADC_ETC->TRIG0_CHAIN_1_0 register.

  NOTE

  The CSEL0 can be omitted if the ADC1 is configured to select the command by software (LPADC1->TCTRLx[CMD_SEL] = 0).

- When the conversion completes, the conversion results are sent from the ADC to active ADC_ETC module trigger channel chain result register. In this case, it is the ADC_ETC->TRIG0_RESULT_1_0 register. The conversion complete signal can also be utilized by the ADC_ETC.

  NOTE

  The ADC_ETC result registers do not accept 13-bit results (only 12-bit results are accepted). This means that if a differential measurement is used for the conversion, it will not accept the sign bit.

- The next trigger is generated depending on the back-to-back mode defined in ADC_ETC->TRIG0_CHAIN_1_0[B2B0].
  - If the B2B mode is disabled, the next trigger is generated based on the delay interval defined in ADC_ETC-> TRIG0_COUNTER [SAMPLE_INTERVAL].

NOTE

the delay interval must be higher than the total conversion time to avoid the next trigger signal generation before the current conversion finishes.

— If the B2B mode is enabled, the next trigger signal is generated immediately after the conversion-complete signal of the previous conversion is generated by the ADC1 module.

- This trigger passes again through the ADC_ETC arbiter and the synchronization module to the ADC1. Now the chain 1 configuration (HTWS1 and CSEL1) defines what trigger and command within the ADC1 module will be used for the conversion.
- Based on the ADC_ETC->TRIG0_CHAIN_1_0[B2B1] configuration, the next trigger is handled.
- The same approach is used until the last chain 7 or until the chain length defined in ADC_ETC->TRIGx[TRIG_CHAIN] field is achieved.

NOTE

Do not configure the active trigger chain ADC trigger selection field HTWSx to 0. This will not initiate the ADC trigger. Hence, no conversion complete signal will be generated and it can cause unpredictable behavior.

NOTE

The active ADC trigger selection field HTWS can only accept one bit set at the once. For example, 0b00000001 represents the ADC trigger 0 selection, 0b00100000 represents the ADC trigger 5, and 0b10000000 represents the ADC trigger 7. Do not use 0b01100000 0b00001110, and so on.

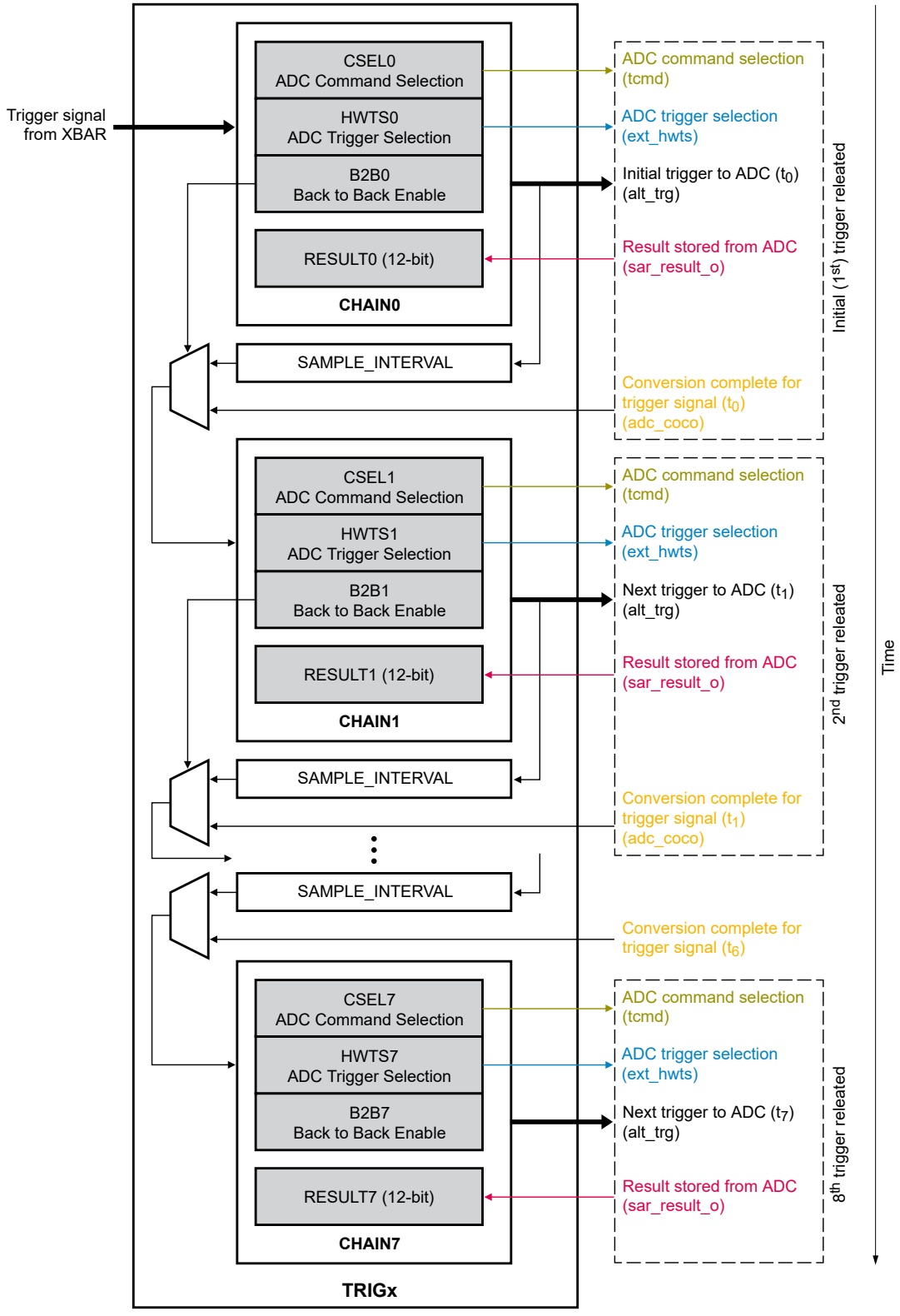

Figure 2. Trigger chaining principle within one ADC_ETC channel trigger

NOTE

The number of trigger signals generated by one channel is defined by the trigger chain length (ADC_ETC->TRIGx_CTRL[TRIG_CHAIN] register). It means that if the length is configured to 3, then the number of total trigger signals generated by the corresponding trigger channel is 4 (including the initial trigger generated by the XBAR signal).

2.3 Analog-to-digital converter (LPADC)

The ADC module used on iMXRT1170 is completely different when compared to the ADC modules used on the previous iMXRT four-digit devices, such as iMXRT1010/RT1020/RT1050/RT1060. It was decided to keep the ADC_ETC module on the iMXRT1170 as the hardware trigger module for the ADC due to compatibility with the previous iMXRT products. To connect the ADC_ETC module with the iMXRT1170-based ADC module, add a wrapper on the ADC module, which allows the interaction between these two modules. The main functionality of the ADC wrapper represents the interface between both ADC modules and the ADC_ETC module to exchange the required information.

It includes the following signals:

- **ext_hwts[0-7]** - selection of the ADC trigger 0-7 by the corresponding ADC_ETC trigger channel (TRIG0-7) chain (CHAIN0-7).
  - **NOTE**: it is represented by 2x8=16 signal lines (2 ADC modules with each ADC module including 8 triggers.
- **tcmd[0-3]** - selection of the ADC trigger command 1-15 by the corresponding ADC_ETC trigger channel (TRIG0-7) chain (CHAIN0-7).
  - **NOTE**: it is represented by 2x4=8 signal lines (2 ADC modules with each ADC module utilizing 4 signal lines for the binary selection of command 1-15).
  - **NOTE**: command 0 represents an invalid selection. Hence, the trigger event is ignored in such configuration.
  - **NOTE**: the tcmd[0-3] signals select the active ADC command only if configured by the ADC. For example, if an active ADC trigger selected by the ***ext_hwts*** signals is 2, then LPADC->TCTRL2[CMD_SEL] selects if a software command will be applied (LPADC->TCTRL2[TCMD]) or if the hardware ***tcmd[0-3]*** signal lines from the ADC_ETC will select the command for this trigger.
- **alt_trg** – active trigger signal line. It can generate only one trigger signal (rising edge) within a specific time interval, based on the arbiter selection.
- **adc_coco[0-7]** – conversion-complete signal generated by the dedicated ADC trigger 0-7.
- **sar_result_o[0-11]** – conversion data results signal lines. They are represented by 12 signal lines for 12-bit data results.
  - **NOTE:** Note that the ADC_ETC module does not consider the 13^{th} sign bit here. It means that using ADC_ETC as a hardware trigger source can handle only 12-bit unsigned result data.

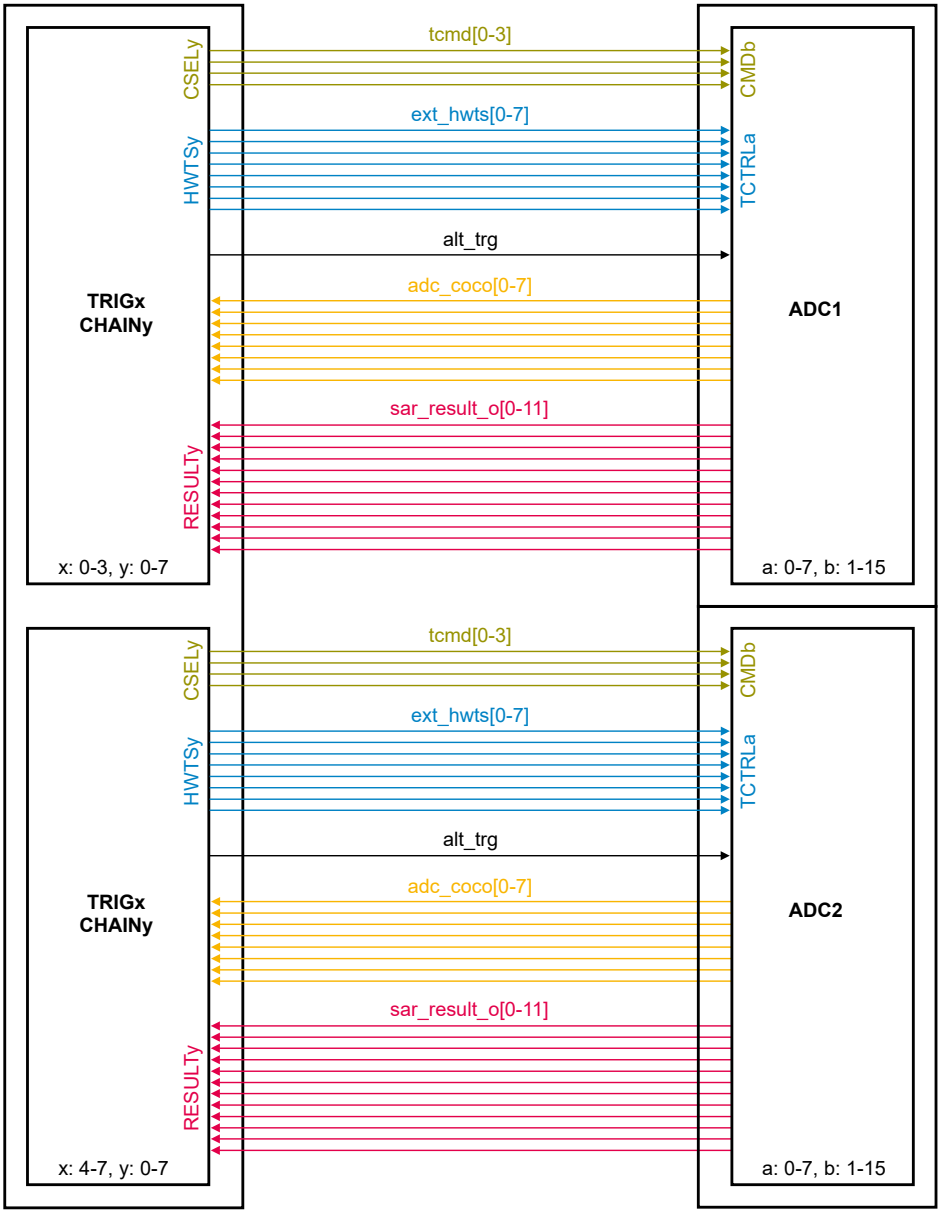

Figure 3. Interface between ADC_ETC and ADCs

The ADC initiates the conversion based on the ADC_ETC trigger signal (rising edge on ***alt_trg***) acceptance if the ADC is configured for a hardware trigger. As mentioned earlier, there are two groups of 4 triggers within the ADC_ETC module. Each group is dedicated for a corresponding ADC module, which means:

- **TRIG0 – TRIG3** within ADC_ETC are dedicated for **ADC1**
- **TRIG4 – TRIG7** within ADC_ETC are dedicated for **ADC2**

NOTE: Do not confuse the two different trigger expressions. The first one is related to trigger channels TRIG0-TRIG7 within the ADC_ETC module, and the second one is related to triggers 0-7 within the ADC modules.

Each trigger channel within one ADC_ETC group generates a trigger signal (***alt_trg***) for the dedicated ADC module based on the ADC_ETC arbiter decision.

There are 8 triggers and 15 commands within one ADC module. The selection of the ADC trigger is based on the ***ext_hwts*** signal generated by the active ADC_ETC trigger channel chain. It is configured in the ADC_ETC-

>TRIGx_CHAIN_y_z[HWTSn] register field. The selection of the ADC command to be used in the conversion is dedicated by the command selection field in the trigger control register of the ADC – LPADCx_TCTRLy[CMD_SEL]. It can select between the following:

— LPADCx_TCTRLy[CMD_SEL] = 0 – command selection by the software configuration of LPADCx_TCTRLy [TCMD].

— LPADCx_TCTRLy[CMD_SEL] = 1 – command selection by the hardware, the ***tcmd*** signal lines from the ADC_ETC selects the command to be used for the ADC conversion.

The active ADC trigger and command defines how the conversion will be processed. After the conversion completes, the result is stored into the ADC FIFO and sent via the ***sar_results_o*** signal lines to the active ADC_ETC trigger channel chain to be stored into its dedicated result register - ADC_ETC->TRIGx_RESULT_y+1_y. The corresponding ADC trigger conversion-complete signal (***adc_coco***) can also be utilized by the ADC_ETC if the back-to-back mode is enabled – ADC_ETC->TRIGx_CHAINy_z[B2Bn] = 1. If it is enabled, the conversion-complete signal (***adc_coco***) initiates the next sequential trigger within the active trigger channel chain.

3 External hardware trigger overview

Figure 4 summarizes the interconnections between all modules required to initiate the ADC conversion by a hardware trigger.

The XBAR selects what peripheral trigger source will be connected to the dedicated trigger channel within the ADC_ETC. Each of the 8 trigger (TRIG0-TRIG7) channels within the ADC_ETC module can generate 8 sequential triggers by chaining (CHAIN0-CHAIN7). Each chain can configure the ADC trigger (trigger 0 to trigger 7 within the ADC module) and command which will be used to control the ADC conversion. The arbiter within the ADC_ETC defines what trigger within the ADC_ETC will control the ADC conversion initiation based on the priority configuration. The synchronizer is used if it is required to trigger both ADC modules simultaneously by one trigger source. Only the active chain within the ADC_ETC trigger channels receives the conversion-complete signal from the ADC module. Based on the conversion-complete signal, it stores the results in the active chain dedicated result register. Each of the ADC modules includes 8 triggers and 15 commands which configure the ADC conversion. When the hardware trigger is selected by the ADC, then the active ADC trigger and the ADC command are specified by the active chain configuration.

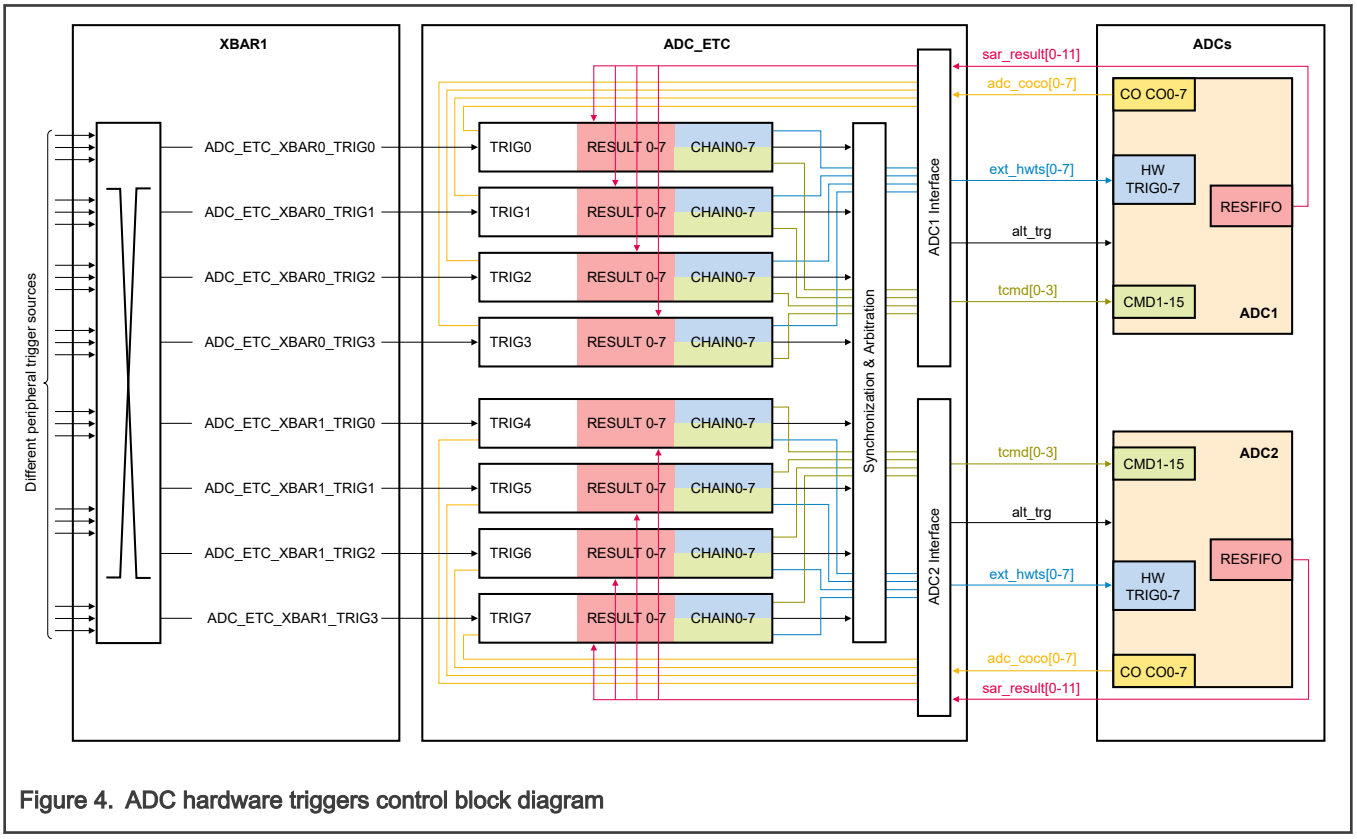

Figure 4. ADC hardware triggers control block diagram

4 Revision history

Table 1. Revision history

Revision number	Date	Substantive changes
0	02/2021	Initial release

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer’s applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 02/2021
Document identifier: AN13097