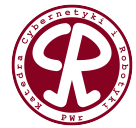
Sterowniki Robotów
Peryferia

Wojciech Domski

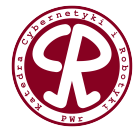
Katedra Cybernetyki i Robotyki,
Politechnika Wrocławska

Wrocław University
of Science and Technology



Plan prezentacji

Wrocław University
of Science and Technology

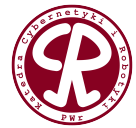


Przykładowe peryferia

Lista najpopularniejszych peryferiów dostępnych w mikrokontrolerach:

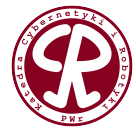
- GPIO,
- ADC,
- DAC,
- timers,
- SPI,
- I2C,
- CAN,
- SDIO,
- USB,
- USART/UART,
- FSMC.

Wrocław University of Science and Technology



Outline

Wrocław University of Science and Technology

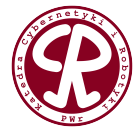


GPIO (1/2)

GPIO jest jednym z najbardziej popularnych peryferiów dostępnych w różnych mikrokontrolerach. Zazwyczaj port cyfrowy może być skonfigurowany w dwóch trybach:

- wejście,
- wyjście.

Wrocław University of Science and Technology

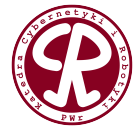


GPIO (2/2)

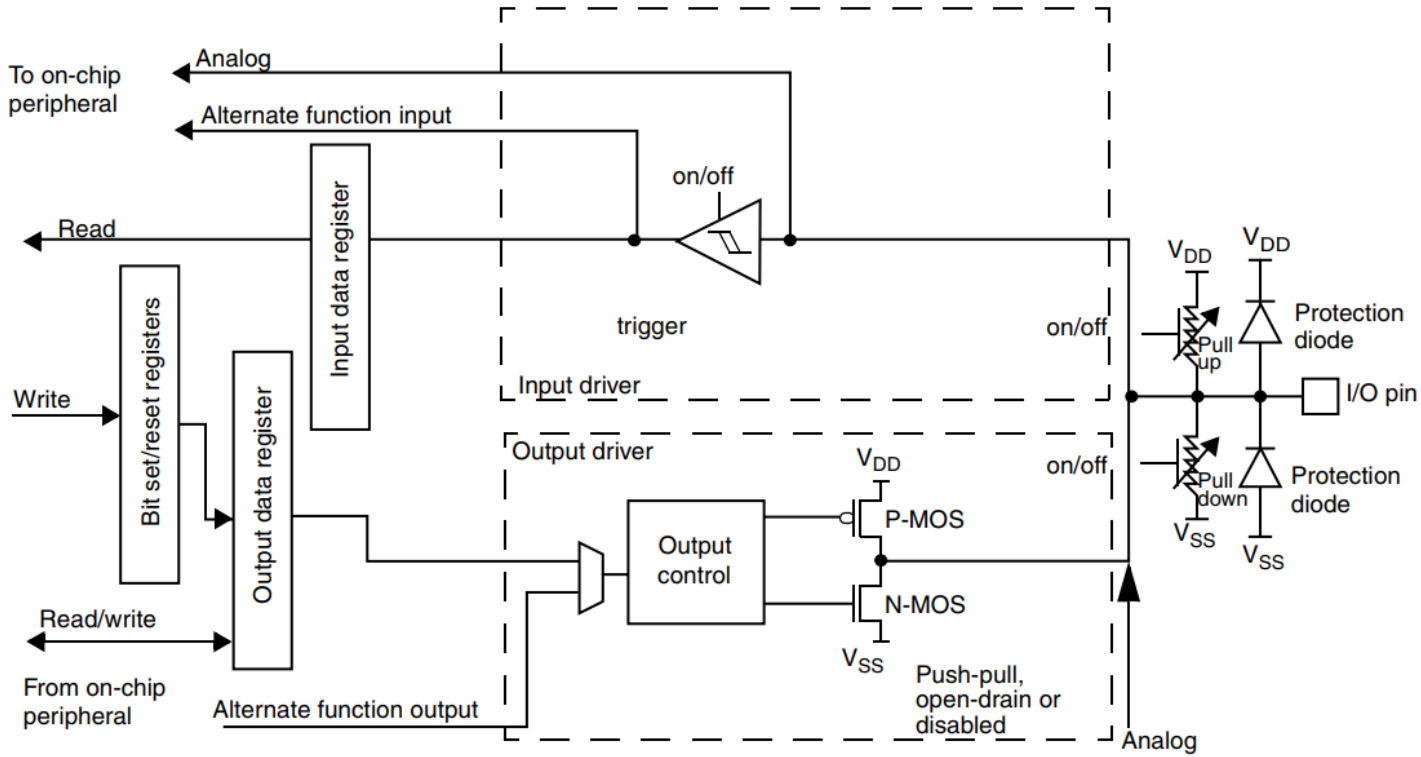
Bardziej rozbudowane peryferia GPIO pozwalają na konfigurację peryferium w innych trybach:

- Input floating,
- Input pull-up,
- Input pull-down,
- Analog,
- Output open-drain with pull-up or pull-down capability,
- Output push-pull with pull-up or pull-down capability,
- Alternate function push-pull with pull-up or pull-down capability,
- Alternate function open-drain with pull-up or pull-down capability.

Wrocław University of Science and Technology

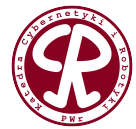


Architektura GPIO



[1]

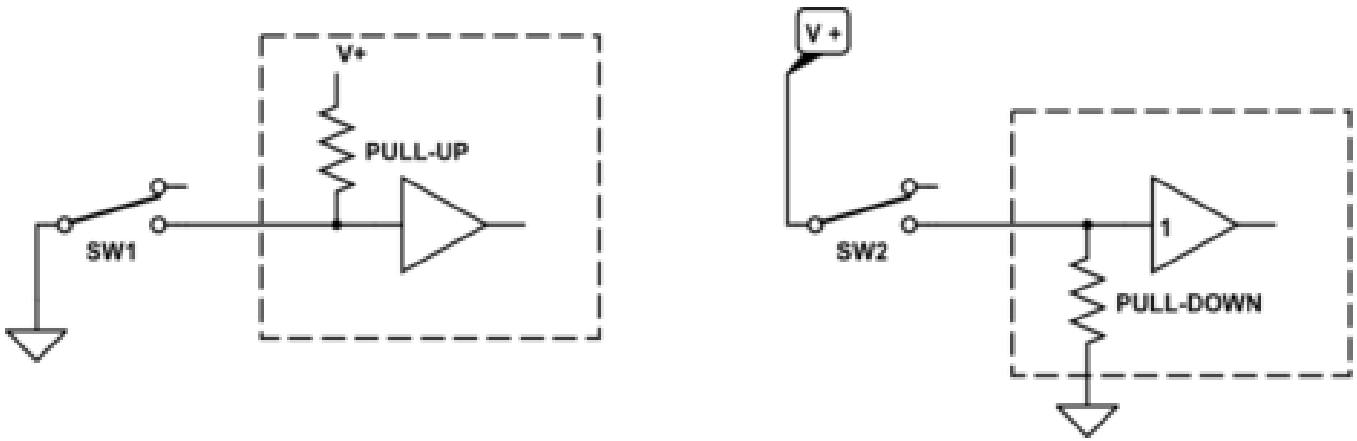
Wrocław University of Science and Technology



[1] http://www.st.com

Cyfrowe wyjście typu open-drain i push-pull (1/6)

Wejścia z rezystorami podciągającymi

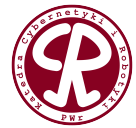


1

Cyfrowe wyjście typu open-drain i push-pull (2/6)

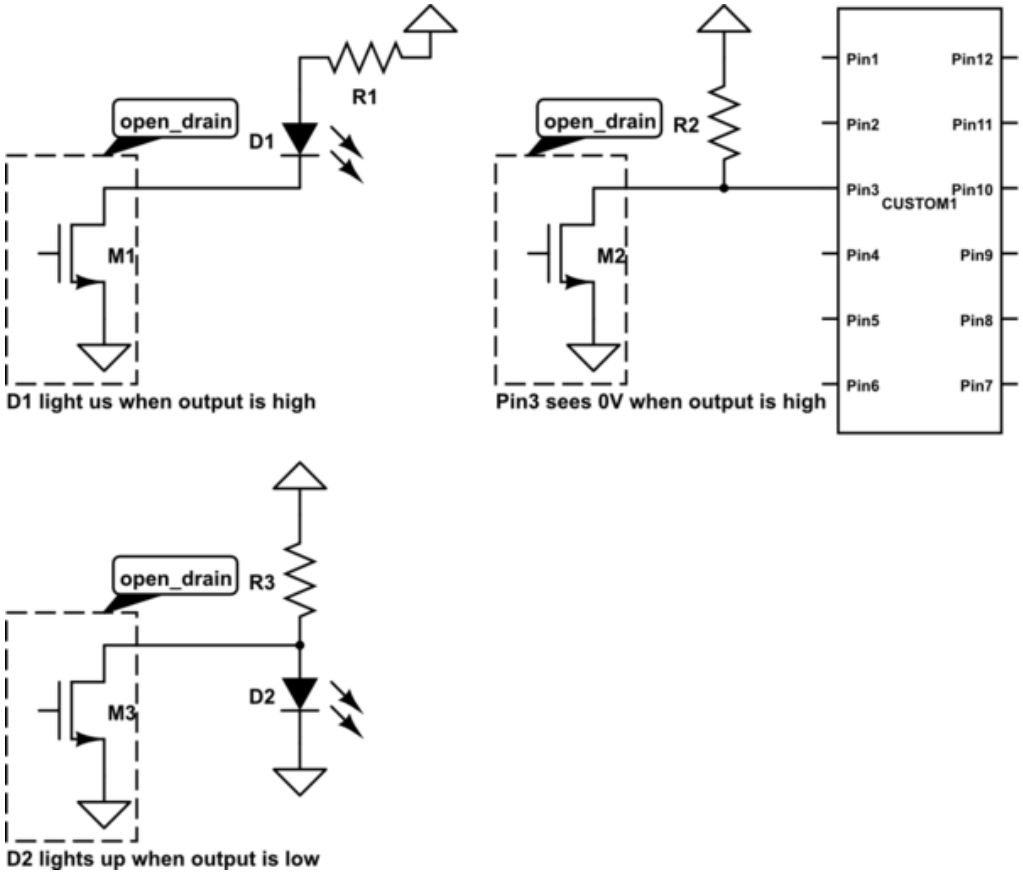
Open-drain (Open-collector) jest typowym układem, który zachowuje się jak przełącznik. Układ ten może być w jednym z dwóch stanów: rozłączonym, bądź podłączonym do masy układu. Zazwyczaj wykorzystuje się transystory NPN bądź tranzystory MOSFET.

Wrocław University of Science and Technology



Cyfrowe wyjście typu open-drain i push-pull (3/6)

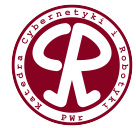
Wyjście Open-drain/Open-collector



Cyfrowe wyjście typu open-drain i push-pull (4/6)

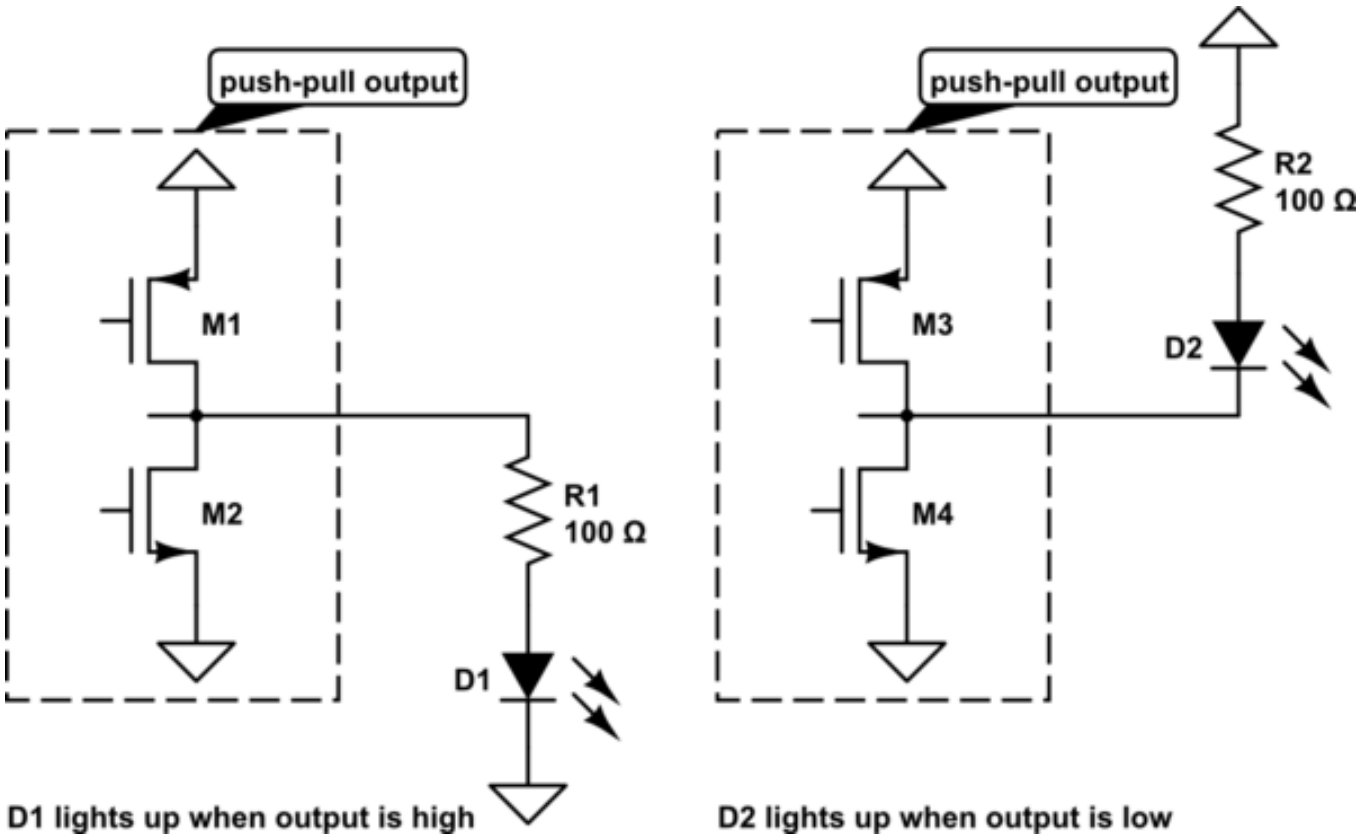
Push-Pull może dostarczać prąd (push), bądź zachowywać się jako absorber prądu (pull). Aby zrealizować tego typu układ wykorzystuje się parę tranzystorów, gdzie w jednym czasie tylko jeden z nich może być aktywny.

Wrocław University
of Science and Technology



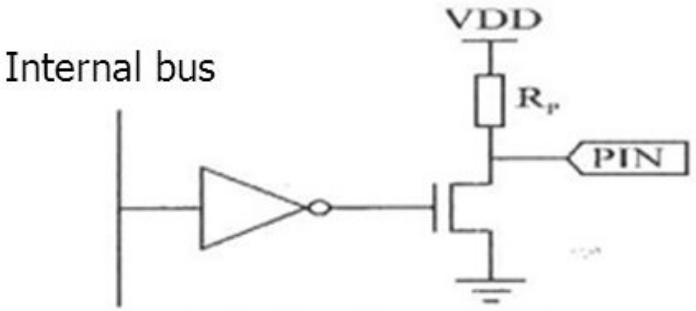
Cyfrowe wyjście typu open-drain i push-pull (5/6)

Wyjście push-pull

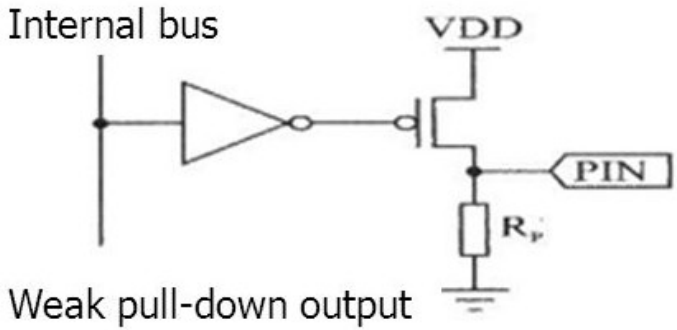


Cyfrowe wyjście typu open-drain i push-pull (6/6)

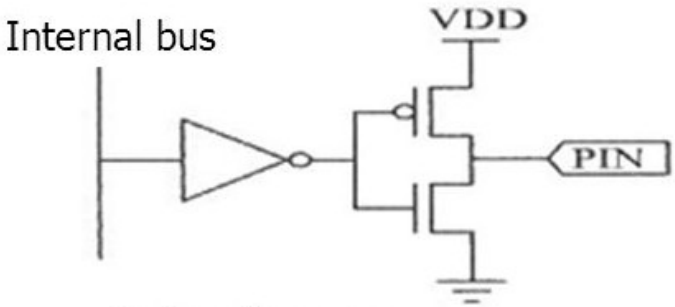
Wyjścia



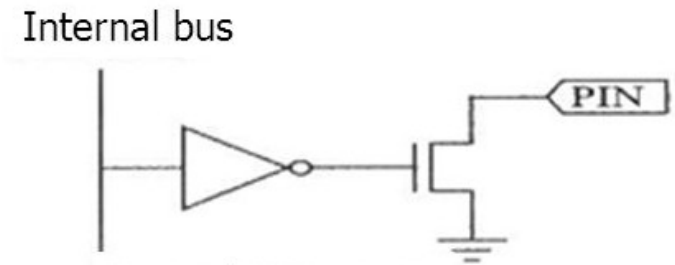
Weak pull-up output



Weak pull-down output



Push-pull output



Open-drain output

2

[1] electronics.stackexchange.com

[2] http://slideplayer.com

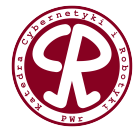
API (1/1)

```
1 HAL_GPIO_ReadPin
2 HAL_GPIO_WritePin
3 HAL_GPIO_TogglePin
4 HAL_GPIO_LockPin
5 HAL_GPIO_EXTI_IRQHandler
6 HAL_GPIO_EXTI_Callback
```

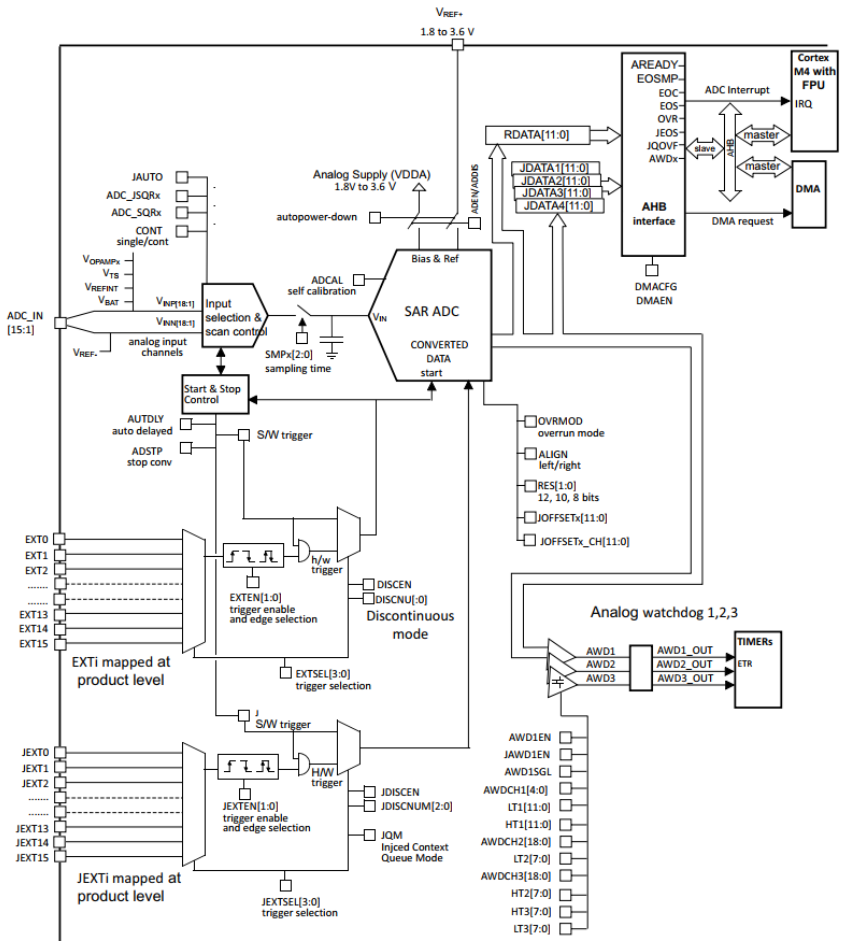
Wrocław University
of Science and Technology

Outline

Wrocław University of Science and Technology



ADC



[2]

Wrocław University of Science and Technology

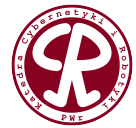
[2] http://www.st.com

Aktywacja konwersji (1/1)

Początek konwersji może być zainicjalizowany przez:

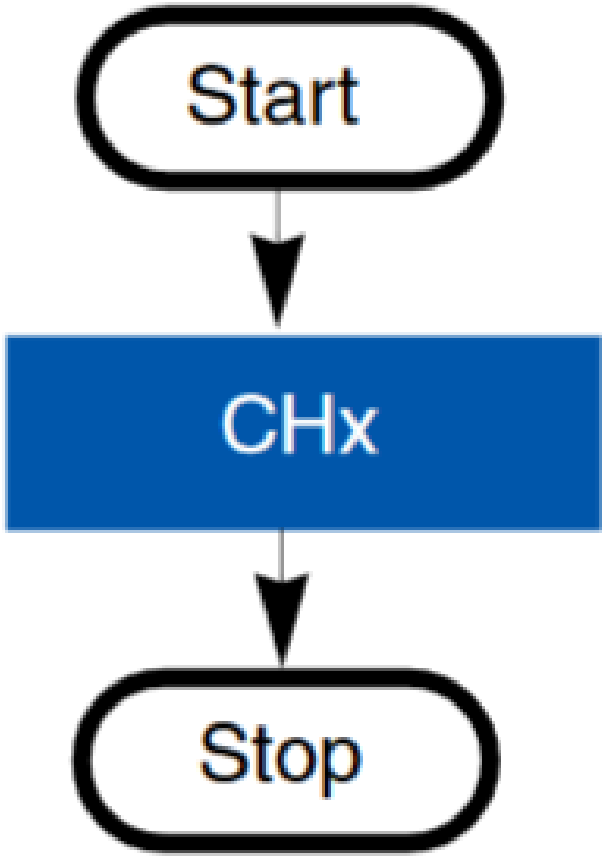
- oprogramowanie, zarówno dla trybu zwykłego, bądź wstrzykiwania (z ang. *injected*),
- sprzętowy wyzwalacz z konfiguracją polaryzacji (wewnętrzny licznik, bądź zdarzenia pochodzące z wejść GPIO) dla trybu zwykłego, bądź wstrzykiwania.

Wrocław University of Science and Technology



Tryby konwersji niezależnej (pojedynczej) (1/6)

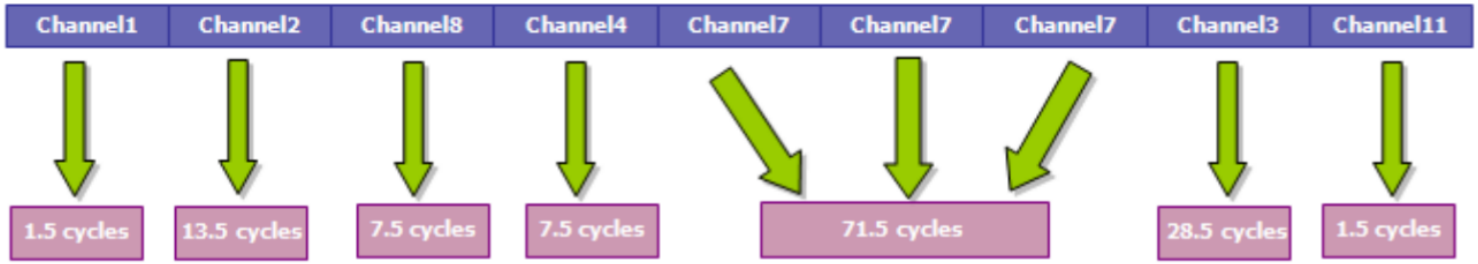
- Single-channel, single conversion mode



1

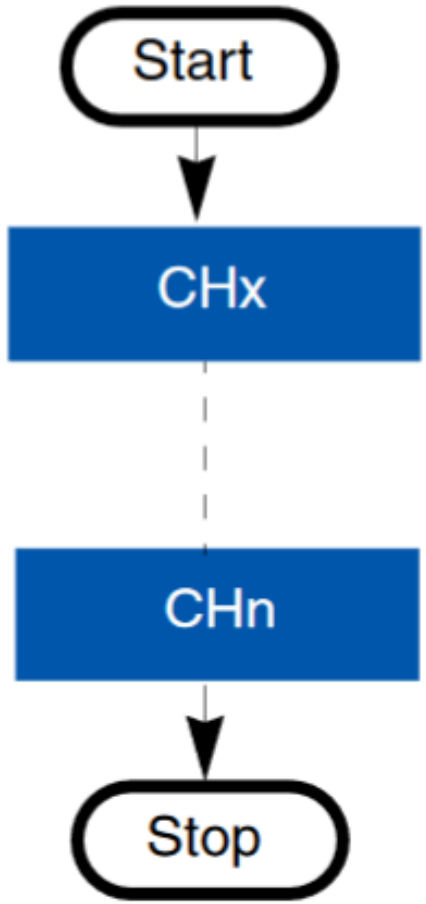
Tryby konwersji niezależnej (pojedynczej) (2/6)

- Multichannel (scan), single conversion mode



1

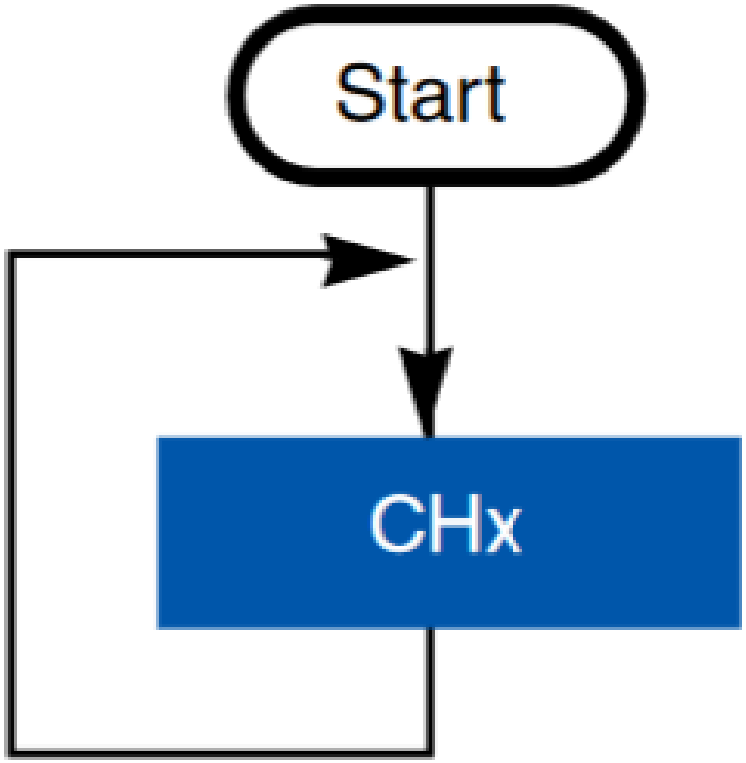
Tryby konwersji niezależnej (pojedynczej) (3/6)



1

Tryby konwersji niezależnej (pojedynczej) (4/6)

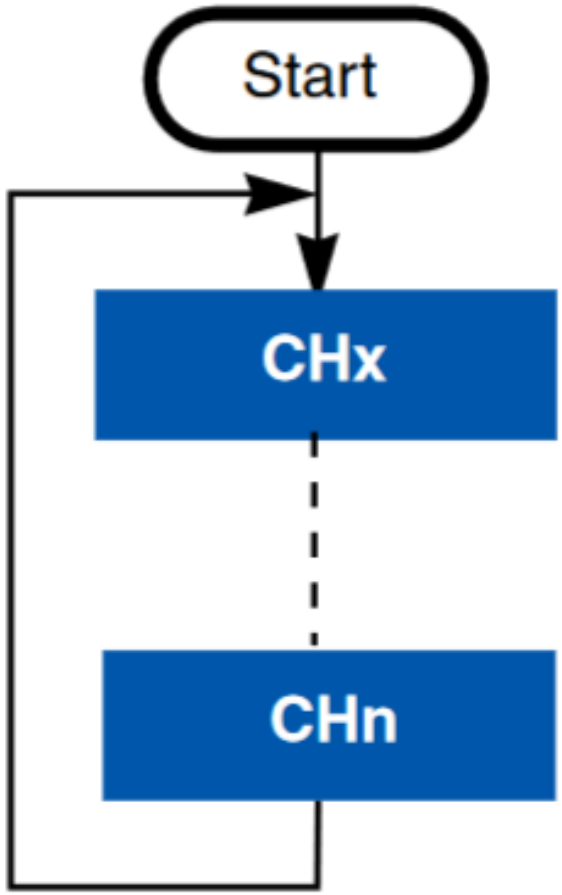
- Single-channel, continuous conversion mode



1

Tryby konwersji niezależnej (pojedynczej) (5/6)

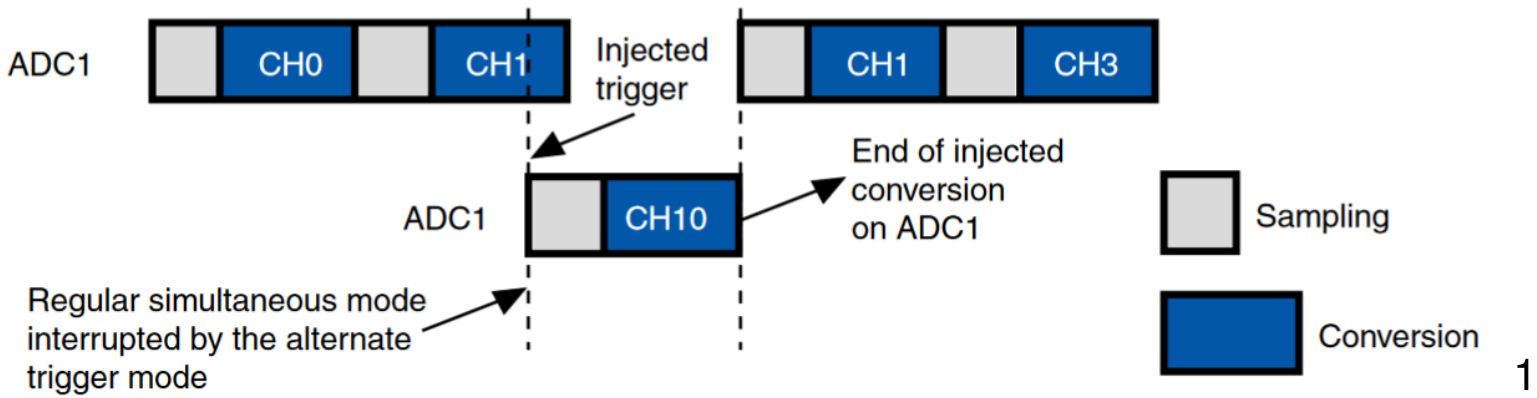
- Multichannel (scan) continuous conversion mode



1

Tryby konwersji niezależnej (pojedynczej) (6/6)

- Injected conversion mode



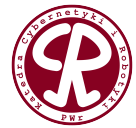
[1]

[1] http://www.st.com

Tryby konwersji synchronicznej (podwójnej) (1/7)

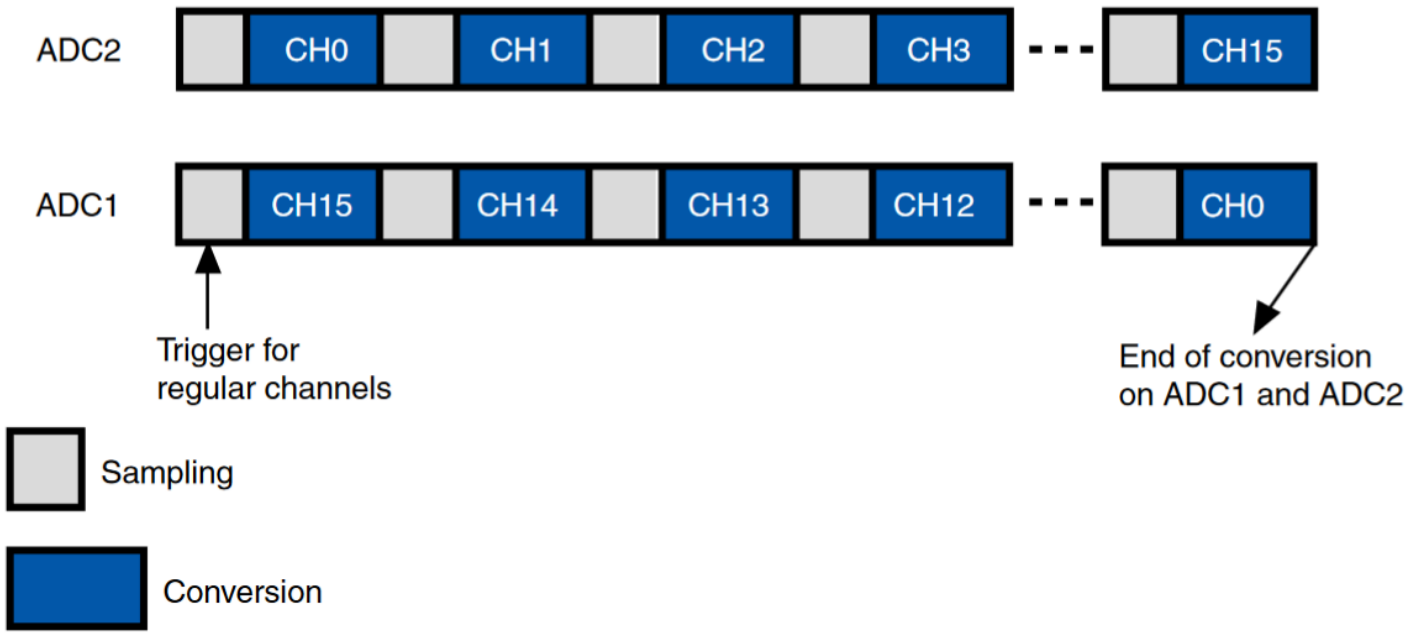
Tryby konwersji podwójnej pozwala na synchronizację dwóch konwerterów ADC, aby było możliwe próbkowanie dwóch kanałów ADC jednocześnie.

Wrocław University
of Science and Technology



Tryby konwersji synchronicznej (podwójnej) (2/7)

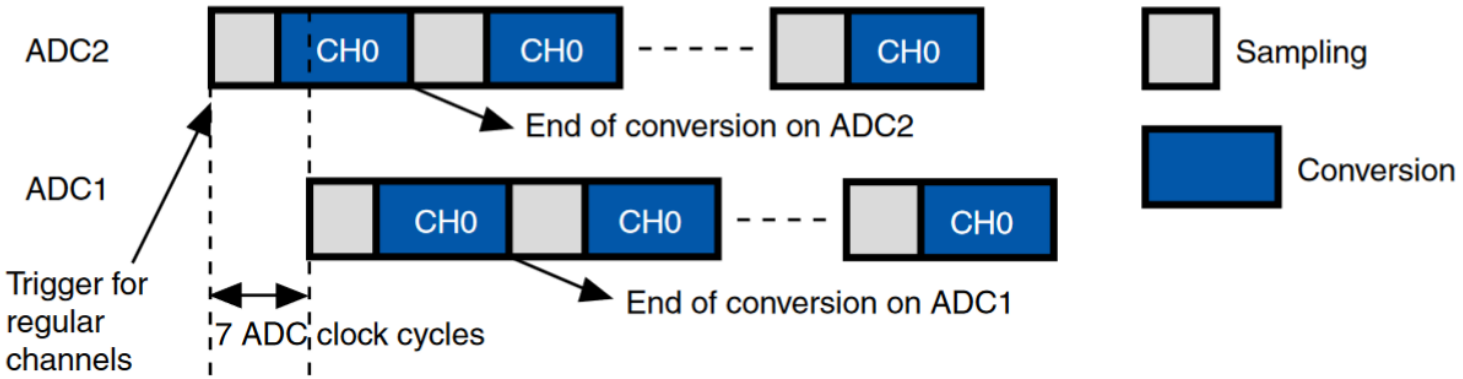
- Dual regular simultaneous mode



1

Tryby konwersji synchronicznej (podwójnej) (3/7)

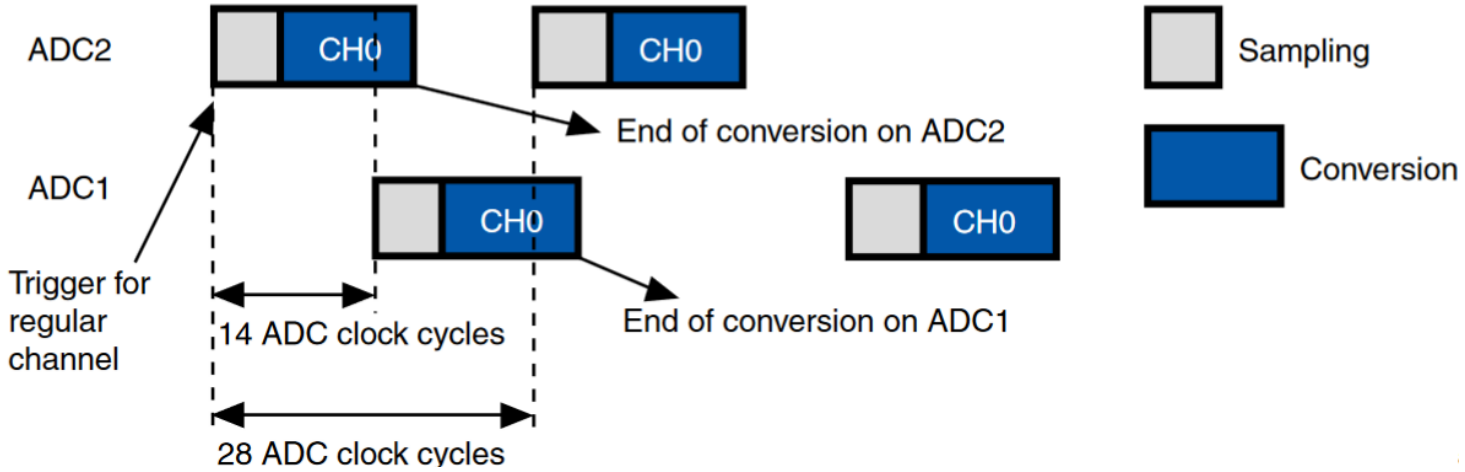
- Dual fast interleaved mode



1

Tryby konwersji synchronicznej (podwójnej) (4/7)

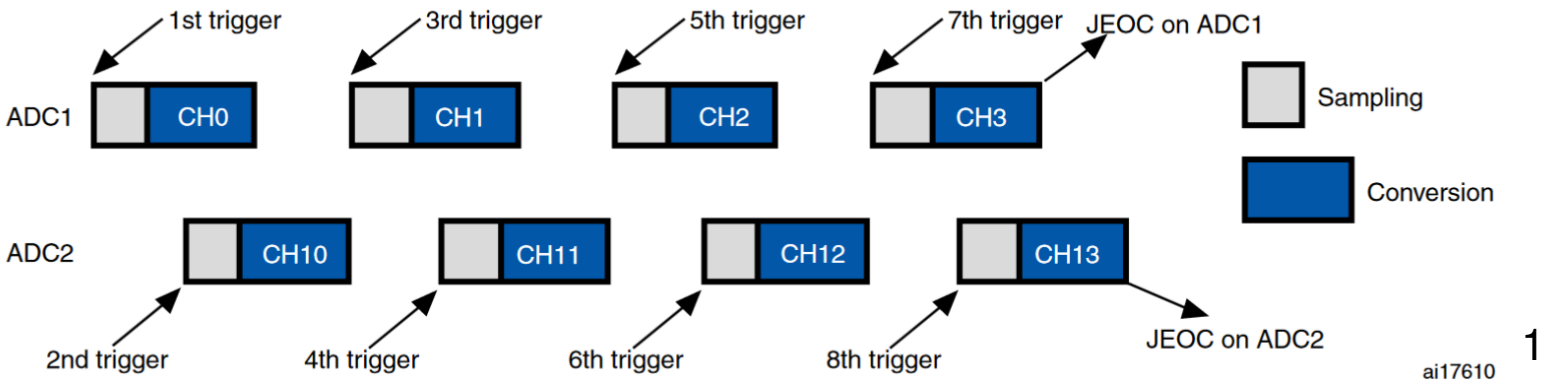
- Dual slow interleaved mode



1

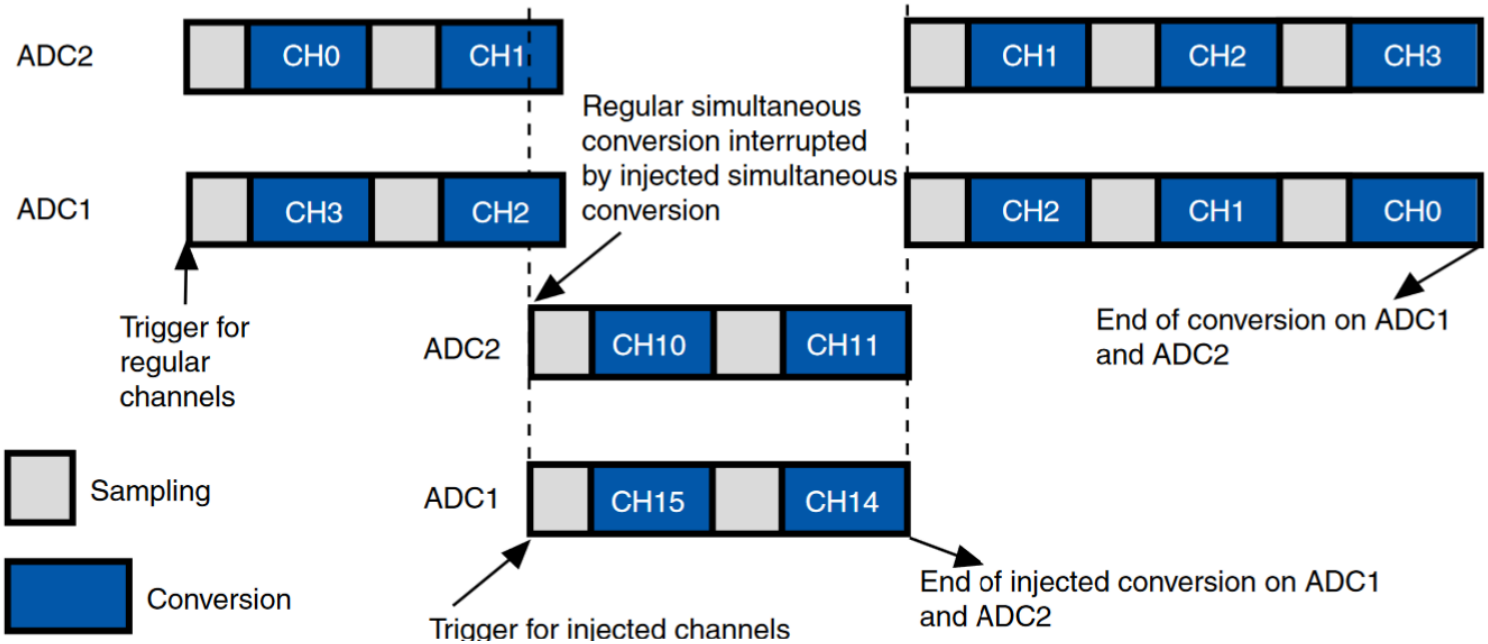
Tryby konwersji synchronicznej (podwójnej) (5/7)

- Dual alternate trigger mode



Tryby konwersji synchronicznej (podwójnej) (6/7)

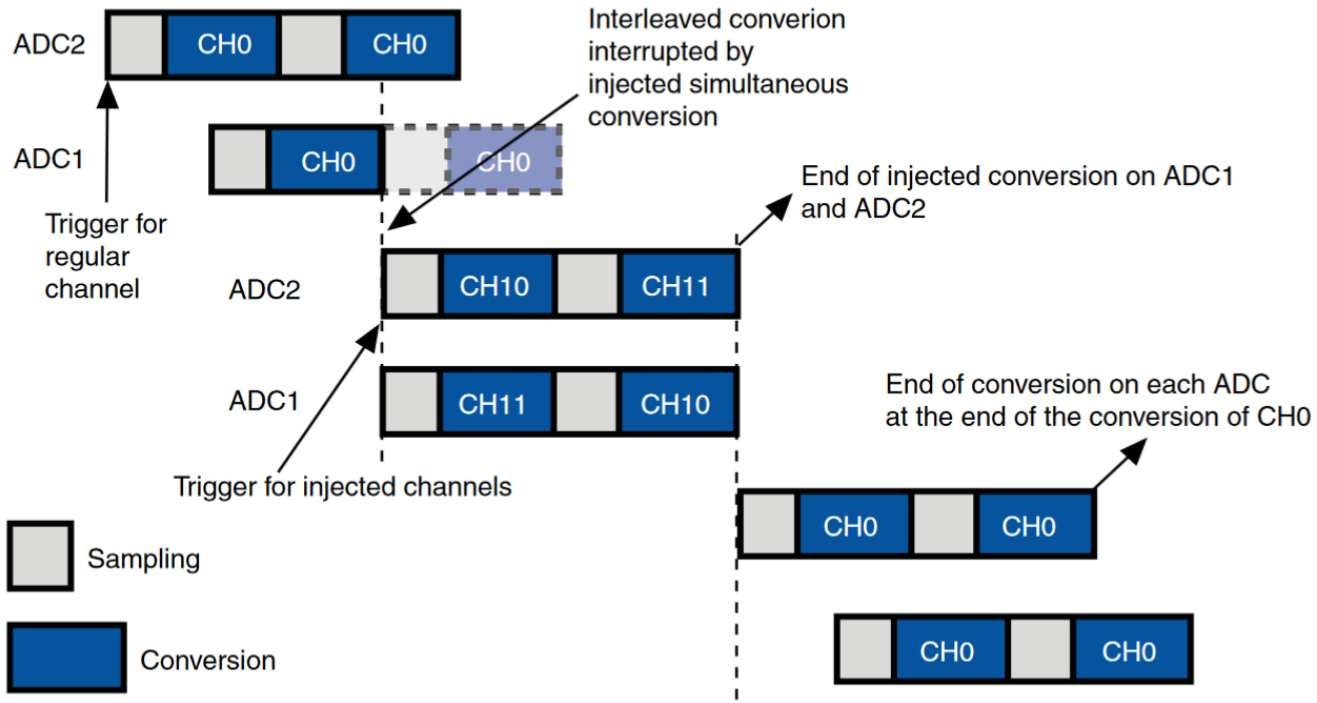
- Dual combined regular/injected simultaneous mode



1

Tryby konwersji synchronicznej (podwójnej) (7/7)

- Dual combined: injected simultaneous + interleaved mode



[1]

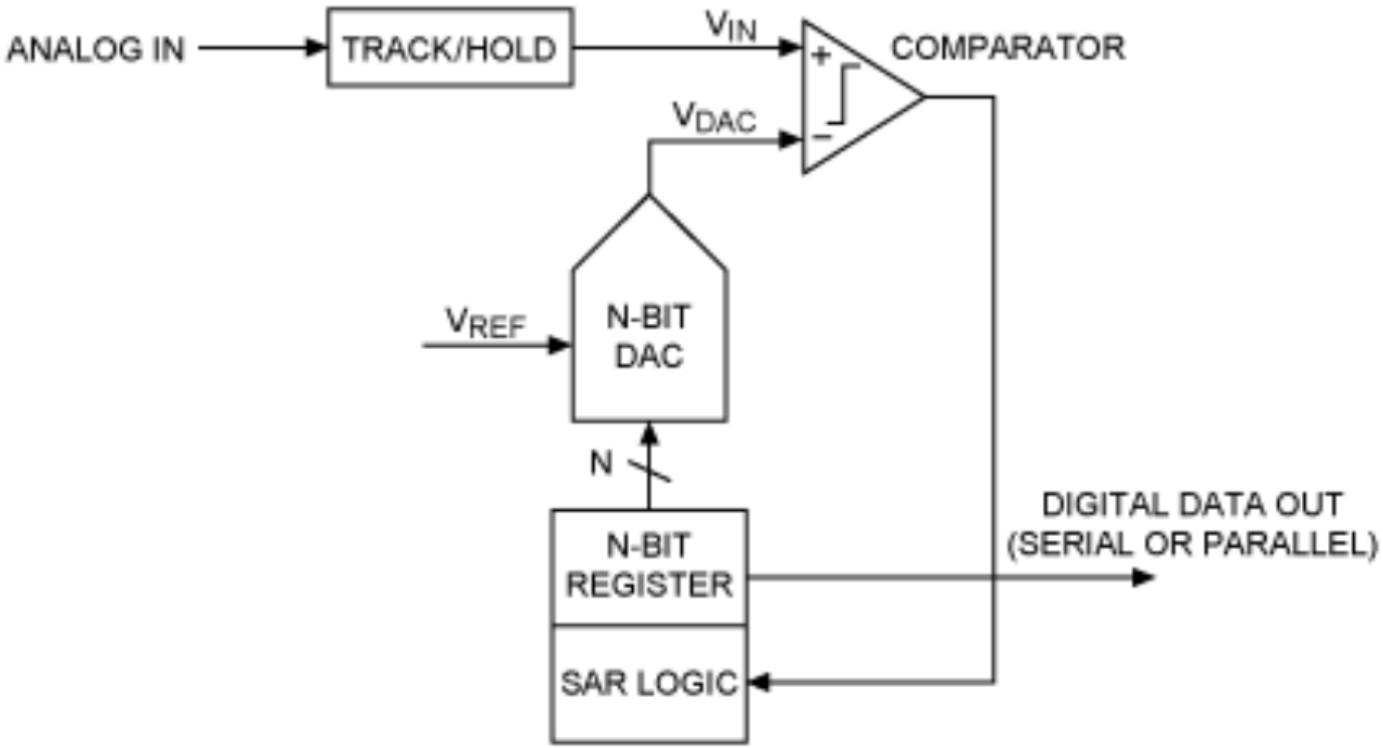
[1] http://www.st.com

Architektury konwerterów ADC

- Successive approximation ADC
- Pipelined ADC
- Flash ADC
- Sigma-Delta converters

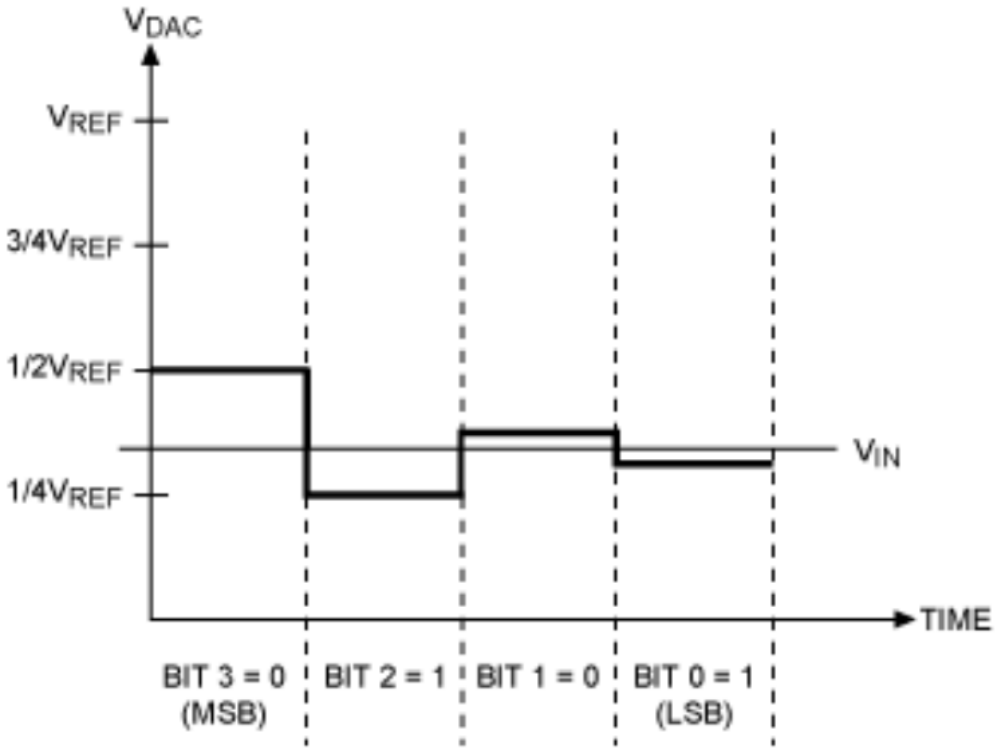
Wrocław University
of Science and Technology

Successive approximation ADC (1/2)



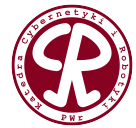
1

Successive approximation ADC (2/2)



[1]

Wrocław University of Science and Technology



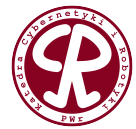
[1] Maxim Integrated, http://www.analog.com

API (1/1)

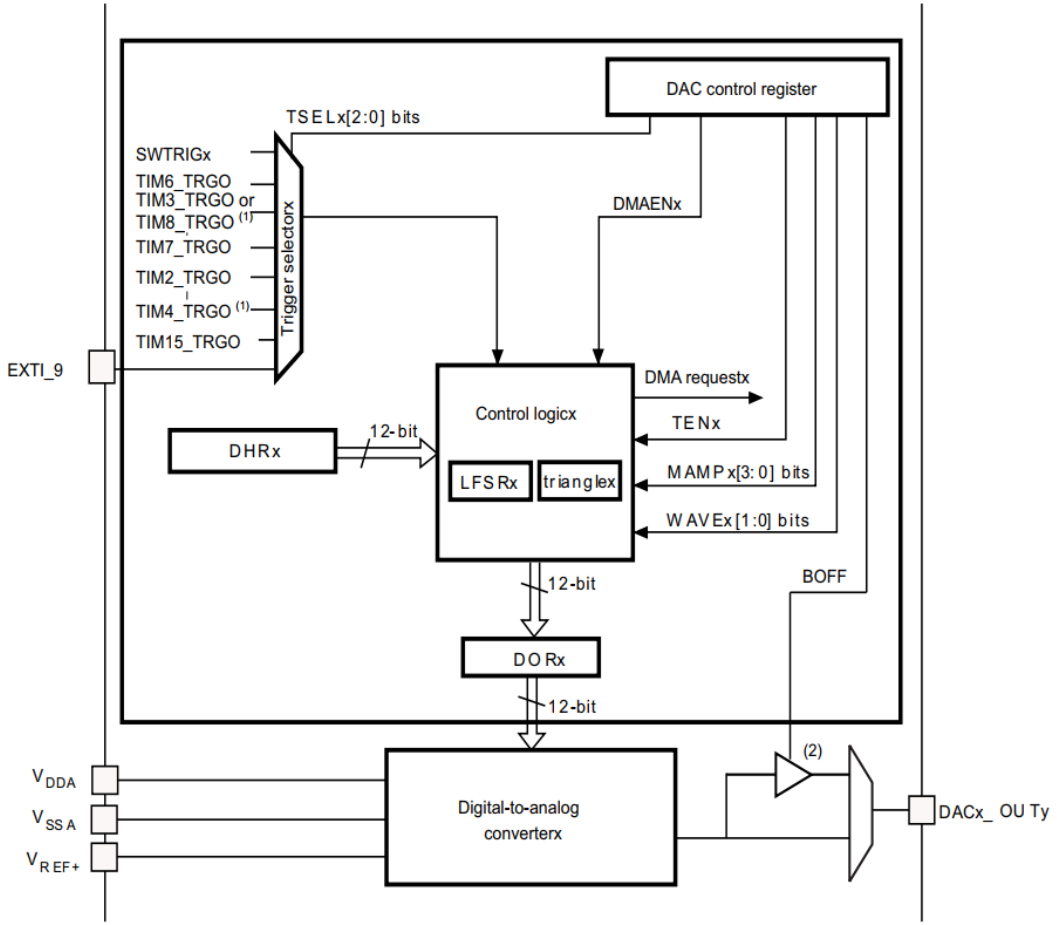
```
1  HAL_ADC_Start
2  HAL_ADC_Stop
3  HAL_ADC_PollForConversion
4  HAL_ADC_PollForEvent
5  HAL_ADC_Start_IT
6  HAL_ADC_Stop_IT
7  HAL_ADC_Start_DMA
8  HAL_ADC_Stop_DMA
9  HAL_ADC_GetValue
10 HAL_ADC_IRQHandler
11 HAL_ADC_ConvCpltCallback
12 HAL_ADC_ConvHalfCpltCallback
13 HAL_ADC_LevelOutOfWindowCallback
14 HAL_ADC_ErrorCallback
15
16 __HAL_ADC_ENABLE()
17 __HAL_ADC_DISABLE();
18 __HAL_ADC_ENABLE_IT()
19 __HAL_ADC_DISABLE_IT();
20 __HAL_ADC_GET_FLAG();
21 __HAL_ADC_GET_IT_SOURCE();
```

Outline

Wrocław University of Science and Technology

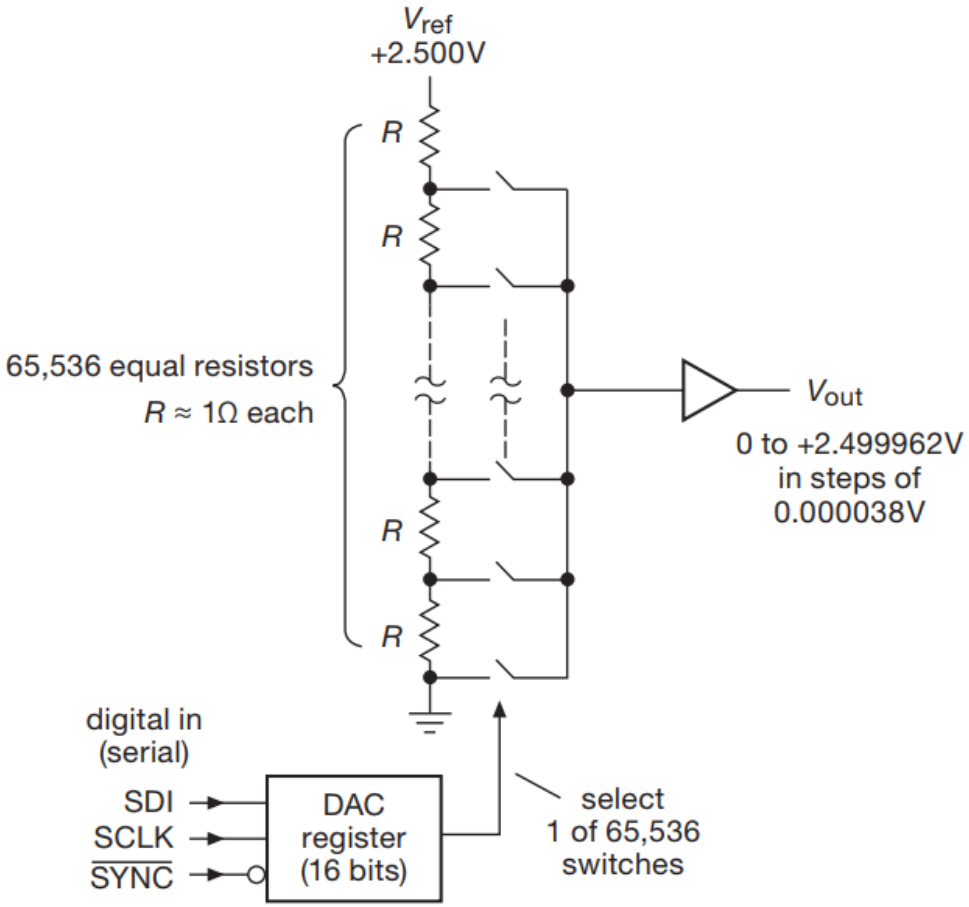


DAC


DAC control register
TSELx[2:0] bits
SWTRIGx
TIM6_TRGO
TIM3_TRGO or
TIM8_TRGO (1)
TIM7_TRGO
TIM2_TRGO
TIM4_TRGO (1)
TIM15_TRGO
Trigger selectorx
EXTI_9
DMAENx
DMA requestx
TENx
MAMPx[3:0] bits
WAVEx[1:0] bits
Control logicx
DHRx
12-bit
LFSRx
triangle x
12-bit
BOFF
DORx
12-bit
(2)
VDDA
VSSA
VREF+
Digital-to-analog
converterx
DACx_OUTy

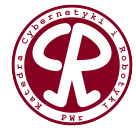
1

Resistor string DAC

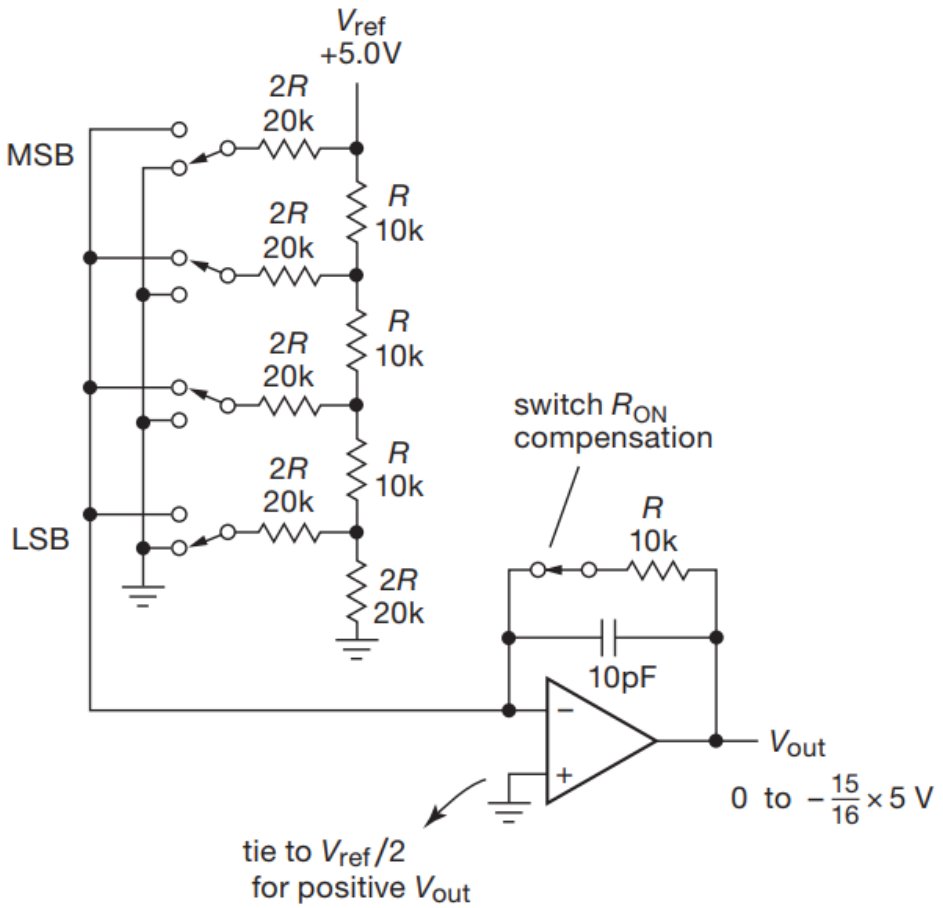


1

Wrocław University
of Science and Technology

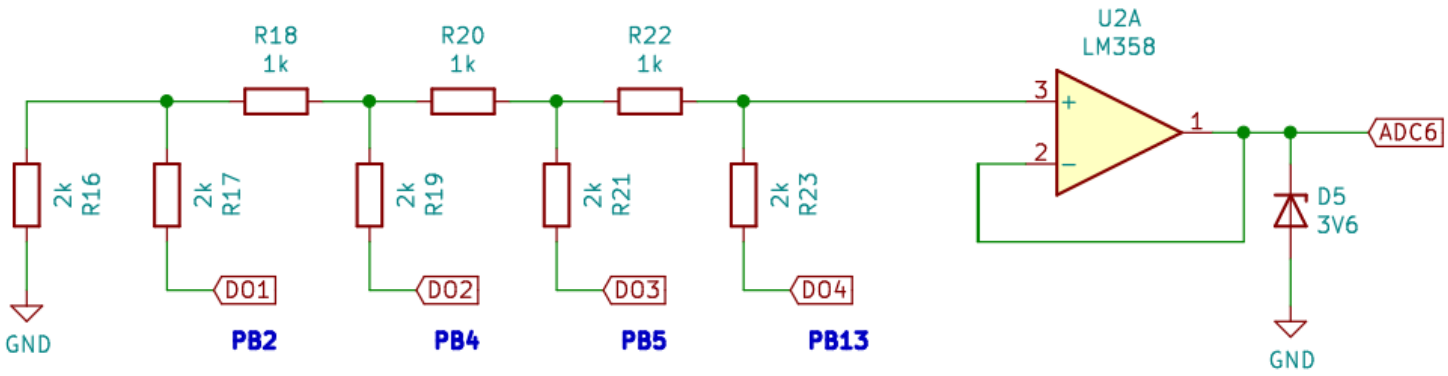


R-2R ladder netowrk DAC (1/2)



1

R-2R ladder netowrk DAC (2/2)



2

Wrocław University of Science and Technology

[1] http://www.st.com

[2] http://domski.pl

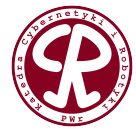
API (1/1)

```
1  HAL_DAC_Start
2  HAL_DAC_Stop
3  HAL_DAC_Start_DMA
4  HAL_DAC_Stop_DMA
5  HAL_DAC_GetValue
6  HAL_DAC_SetValue
7  HAL_DAC_ConvCpltCallbackCh1
8  HAL_DAC_ConvHalfCpltCallbackCh1
9  HAL_DAC_ErrorCallbackCh1
10 HAL_DAC_DMAUnderrunCallbackCh1
11 HAL_DAC_ConfigChannel
12 HAL_DAC_IRQHandler
13
14 __HAL_DAC_ENABLE()
15 __HAL_DAC_DISABLE();
16 __HAL_DAC_ENABLE_IT()
17 __HAL_DAC_DISABLE_IT();
18 __HAL_DAC_GET_FLAG();
19 __HAL_DAC_GET_IT_SOURCE();
```

Wrocław University of Science and Technology

Outline

Wrocław University of Science and Technology



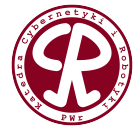
Liczniki

Można wyróżnić **trzy** podstawowe tryby układu liczącego:

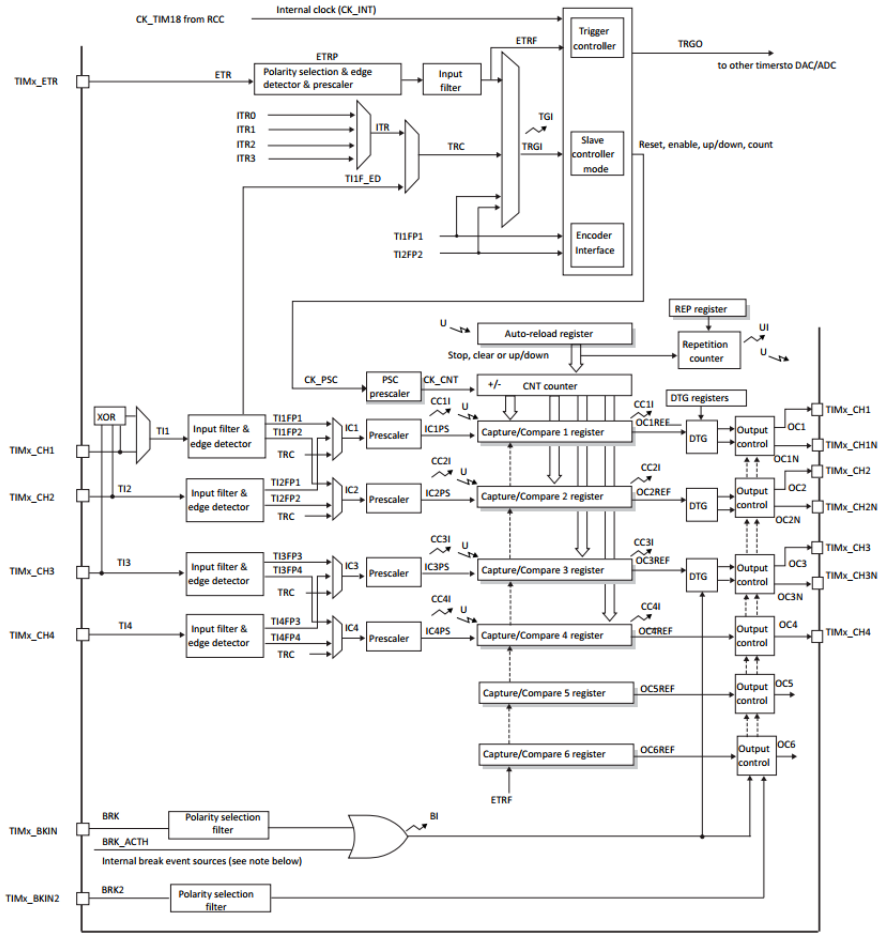
- Generator podstawy czasu (z ang. *time base*),
- Output Compare (OC), np. Pulse Width Modulation (PWM),
- Input Capture (IC).

Zazwyczaj mikrokontroler dostarcza innych trybów konfiguracji licznika, które pozwalają na realizację wyspecjalizowanych systemów.
Przykładowo licznik może być skonfigurowany jako wejście enkodera kwadraturowego, czy pomiar parametrów sygnału PWM. Możliwe jest również generowanie sygnałów cyfrowych przy wykorzystaniu układu liczącego wraz z kontrolerem DMA.

Wrocław University
of Science and Technology



Architektura



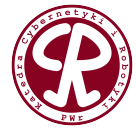
[3]

[3] http://www.st.com

Rodzaje pinów układu liczącego

- TIMx_CHy – y^{wy} kanał wejścia/wyjścia,
- TIMx_CHyN – y^{wy} wyjście komplementarne,
- TIMx_ETR – zewnętrzne wejście wyzwalacza. Może być wykorzystane jako zewnętrzny sygnał taktujący, wyzwalacz, punkt łączący liczniki w łańcuch master-slave, czy wejście resetujące sygnału PWM,
- TIMx_BKINy – wejście przerywające (z ang. *break input*) wykorzystywane do ochrony końcówek mocy w przypadku sterowania końcówkami mocy np. dla silnika DC.

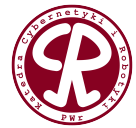
Wrocław University of Science and Technology



Rejestry układów liczących

- Counter register (TIMx_CNT) – przechowuje obecny stan licznika (liczbę jego taktów),
- Prescaler register (TIMx_PSC) – pozwala na przeskalowanie częstotliwości układu liczącego w dół,
- Auto-reload register (TIMx_ARR) – przechowuje wartość graniczną licznika np. w przypadku generowania sygnału PWM pozwala na określenie okresu,
- Repetition counter register (TIMx_RCR) – określa liczbę powtórzeń pełnych cykli licznika, np. wykorzystywany do generowania sygnału DShot.

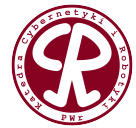
Wrocław University of Science and Technology



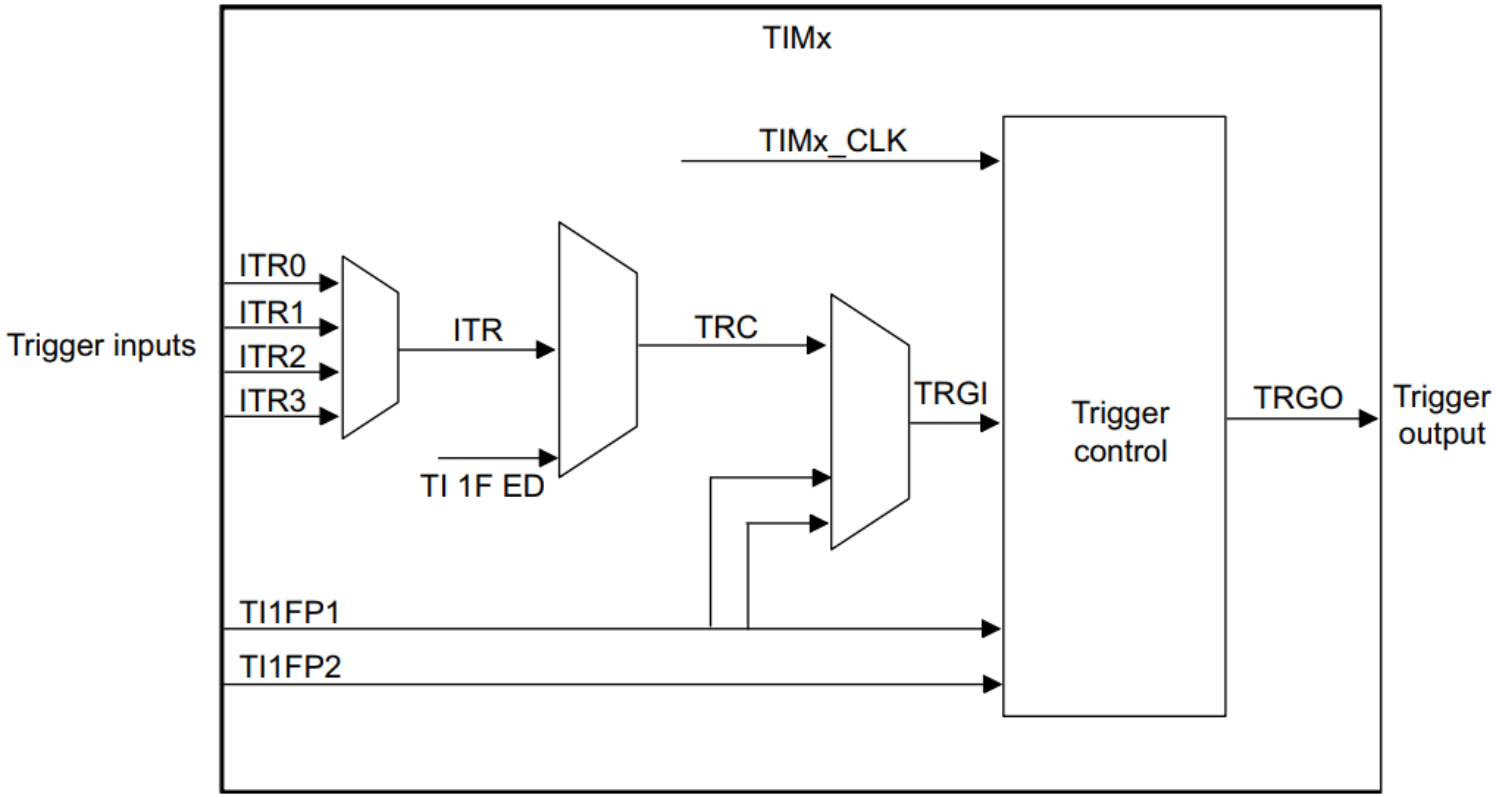
Łączenie układów liczących (z ang. *Timer chaining*) (1/8)

Liczniki mogą być ze sobą połączone w trybie master/slave. Pozwala to na „napędzanie” jednego licznika sygnałem pochodzącym z drugiego układu liczącego.
Najprostszą aplikacją pozwalającą na połączenie dwóch liczników ze sobą jest generator podstawy czasu o wydłużonym okresie. Zaletą tego połączenie jest wydłużenie maksymalnego czasu, po którym licznik się zresetuje przy jednoczesnym zachowaniu małego ziarna.
Możliwe jest połączenie typu 1-n, tzn. jeden licznik nadrzędny (z ang. *master*), może być połączony do n liczników podrzędnych (z ang. *slave*).

Wrocław University of Science and Technology



Łączenie układów liczących (z ang. *Timer chaining*) (2/8)



4

Łączenie układów liczących (z ang. *Timer chaining*) (3/8)

Konfiguracja w trybie nadrzędnym (z ang. *master*).
Wyjście wyzwalacza (z ang. *trigger output, (TRGO)*) może być wybrane jako jedno z poniższych:

- Reset – bit UG z rejestru TIMx_EGR wyzwala sygnał na wyjściu TRGO,
- Enable – sygnał załączenie licznika przekazywany jest na wyjście TRGO w celu uruchomienia kilku liczników w tym samym czasie, bądź kontroli okna aktywności liczników podrzędnych,
- Update – wyjscie TRGO wyzwalane jest zdarzeniem aktualizacji; przykładowo urządzenie nadrzędne może być wykorzystane jako prescaler dla podrzędnego układu liczącego.

Wrocław University of Science and Technology

Łączenie układów liczących (z ang. *Timer chaining*) (4/8)

- Compare pulse – wyjście licznika wysyła zbocze narastające wtedy gdy flaga CC1IF ma zostać ustawiona (nawet w przypadku, gdy jest już ustawiona) jeśli pojawi się zdarzenie typu capture, bądź compare match,
- OCxRef – sygnał OCxREF jest wykorzystywany jako wyjście TRGO, $x \in 1, 2, 3, 4$.

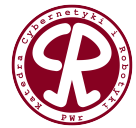
Wrocław University of Science and Technology

Łączenie układów liczących (z ang. *Timer chaining*) (5/8)

Konfiguracja licznika w trybie nadrzędnym wymaga:

1. Ustawienia parametrów pracy licznika.
2. Wybranie wyjścia wyzwalającego poprzez ustawienie bitów MMS (Master Mode Selection) w rejestrze konfiguracyjnym TIMx_CR2.
3. Ustawieniu bitu MSM (Master/Slave Mode) w rejestrze SMCR w celu synchronizacji pomiędzy obecnym licznikiem (nadrzędnym), a licznikami podrzędnymi poprzez wyjście TRGO.

Wrocław University of Science and Technology

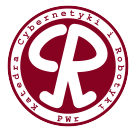


Łączenie układów liczących (z ang. *Timer chaining*) (6/8)

Konfiguracja w trybie podrzędnym (z ang. *slave*).
Podrzędnym układ liczący połączony jest z nadrzędnym układem liczącym poprzez tzw. linię wejściową (z ang. *input trigger, ITR*). Każda linia ITR jest podłączona wewnętrznie do danego licznika.
tryby pracy liczniki w konfiguracji podrzędnej:

- Reset mode – zbocze narastające na wejściu TRGI reinicjalizuje licznik i wywołuje aktualizację rejestrów,
- Gated mode – zegar licznika jest włączony, gdy wejście TRGI jest w stanie wysokim. Licznik zatrzymuje się (bez resetowania stanu), gdy wejście TRGI jest w stanie niskim. Zatem, start i zatrzymanie licznika jest kontrolowane,

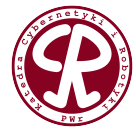
Wrocław University of Science and Technology



Łączenie układów liczących (z ang. *Timer chaining*) (7/8)

- Trigger mode – licznik zostaje uruchomiony gdy na wejściu TRGI pojawi się zbocze narastające. Stan licznika pozostaje bez zmian. Kontrolowany jest wyłącznie uruchomienie licznika,
- External clock mode 1 – narastające zbocze na wybranym wejściu TRGI napędza licznik,
- Combined reset + trigger mode – narastające zbocze wybranego wejścia TRGI reinicjalizuje stan licznika, generuje zdarzenie aktualizacji rejestrów oraz uruchamia licznik. Dostępność tego trybu jest ograniczona.

Wrocław University of Science and Technology



Łączenie układów liczących (z ang. *Timer chaining*) (8/8)

Aby skonfigurować licznik w trybie slave należy:

1. Wybrać właściwy tryb pracy licznika poprzez zapis bitów SMS (Slave Mode Selection) w rejestrze SMCR.
2. Wybrać wyzwalanie wewnętrzne poprzez ustawienie bitów TS (z ang. *trigger selection*) w rejestrze SMCR.

Wrocław University of Science and Technology

[4] http://www.st.com

API (1/5)

```
1  HAL_TIM_Base_Init
2  HAL_TIM_Base_DeInit
3  HAL_TIM_Base_MspInit
4  HAL_TIM_Base_MspDeInit
5  HAL_TIM_Base_Start
6  HAL_TIM_Base_Stop
7  HAL_TIM_Base_Start_IT
8  HAL_TIM_Base_Stop_IT
9  HAL_TIM_Base_Start_DMA
10 HAL_TIM_Base_Stop_DMA
11
12 HAL_TIM_OC_Init
13 HAL_TIM_OC_DeInit
14 HAL_TIM_OC_MspInit
15 HAL_TIM_OC_MspDeInit
16 HAL_TIM_OC_Start
17 HAL_TIM_OC_Stop
18 HAL_TIM_OC_Start_IT
19 HAL_TIM_OC_Stop_IT
20 HAL_TIM_OC_Start_DMA
21 HAL_TIM_OC_Stop_DMA
22
```

API (2/5)

```
23 HAL_TIM_PWM_Init
24 HAL_TIM_PWM_DeInit
25 HAL_TIM_PWM_MspInit
26 HAL_TIM_PWM_MspDeInit
27 HAL_TIM_PWM_Start
28 HAL_TIM_PWM_Stop
29 HAL_TIM_PWM_Start_IT
30 HAL_TIM_PWM_Stop_IT
31 HAL_TIM_PWM_Start_DMA
32 HAL_TIM_PWM_Stop_DMA
33
34 HAL_TIM_IC_Init
35 HAL_TIM_IC_DeInit
36 HAL_TIM_IC_MspInit
37 HAL_TIM_IC_MspDeInit
38 HAL_TIM_IC_Start
39 HAL_TIM_IC_Stop
40 HAL_TIM_IC_Start_IT
41 HAL_TIM_IC_Stop_IT
42 HAL_TIM_IC_Start_DMA
43 HAL_TIM_IC_Stop_DMA
44
```

API (3/5)

```
45 HAL_TIM_OnePulse_Init
46 HAL_TIM_OnePulse_DeInit
47 HAL_TIM_OnePulse_MspInit
48 HAL_TIM_OnePulse_MspDeInit
49 HAL_TIM_OnePulse_Start
50 HAL_TIM_OnePulse_Stop
51 HAL_TIM_OnePulse_Start_IT
52 HAL_TIM_OnePulse_Stop_IT
53
54 HAL_TIM_Encoder_Init
55 HAL_TIM_Encoder_DeInit
56 HAL_TIM_Encoder_MspInit
57 HAL_TIM_Encoder_MspDeInit
58 HAL_TIM_Encoder_Start
59 HAL_TIM_Encoder_Stop
60 HAL_TIM_Encoder_Start_IT
61 HAL_TIM_Encoder_Stop_IT
62 HAL_TIM_Encoder_Start_DMA
63 HAL_TIM_Encoder_Stop_DMA
64
65 HAL_TIM_PeriodElapsedCallback
66 HAL_TIM_PeriodElapsedHalfCpltCallback
```

API (4/5)

```
67 HAL_TIM_OC_DelayElapsedCallback
68 HAL_TIM_IC_CaptureCallback
69 HAL_TIM_IC_CaptureHalfCpltCallback
70 HAL_TIM_PWM_PulseFinishedCallback
71 HAL_TIM_PWM_PulseFinishedHalfCpltCallback
72 HAL_TIM_TriggerCallback
73 HAL_TIM_TriggerHalfCpltCallback
74 HAL_TIM_ErrorCallback
75
76 __HAL_TIM_ENABLE()
77 __HAL_TIM_DISABLE();
78 __HAL_TIM_ENABLE_IT()
79 __HAL_TIM_DISABLE_IT();
80 __HAL_TIM_GET_FLAG();
81 __HAL_TIM_GET_IT_SOURCE();
82
83 __HAL_TIM_IS_TIM_COUNTING_DOWN
84 __HAL_TIM_SET_PRESCALER
85 __HAL_TIM_SET_COUNTER
86 __HAL_TIM_GET_COUNTER
87 __HAL_TIM_SET_AUTORELOAD
88 __HAL_TIM_GET_AUTORELOAD
```

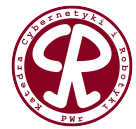
API (5/5)

89 __HAL_TIM_SET_COMPARE
90 __HAL_TIM_GET_COMPARE

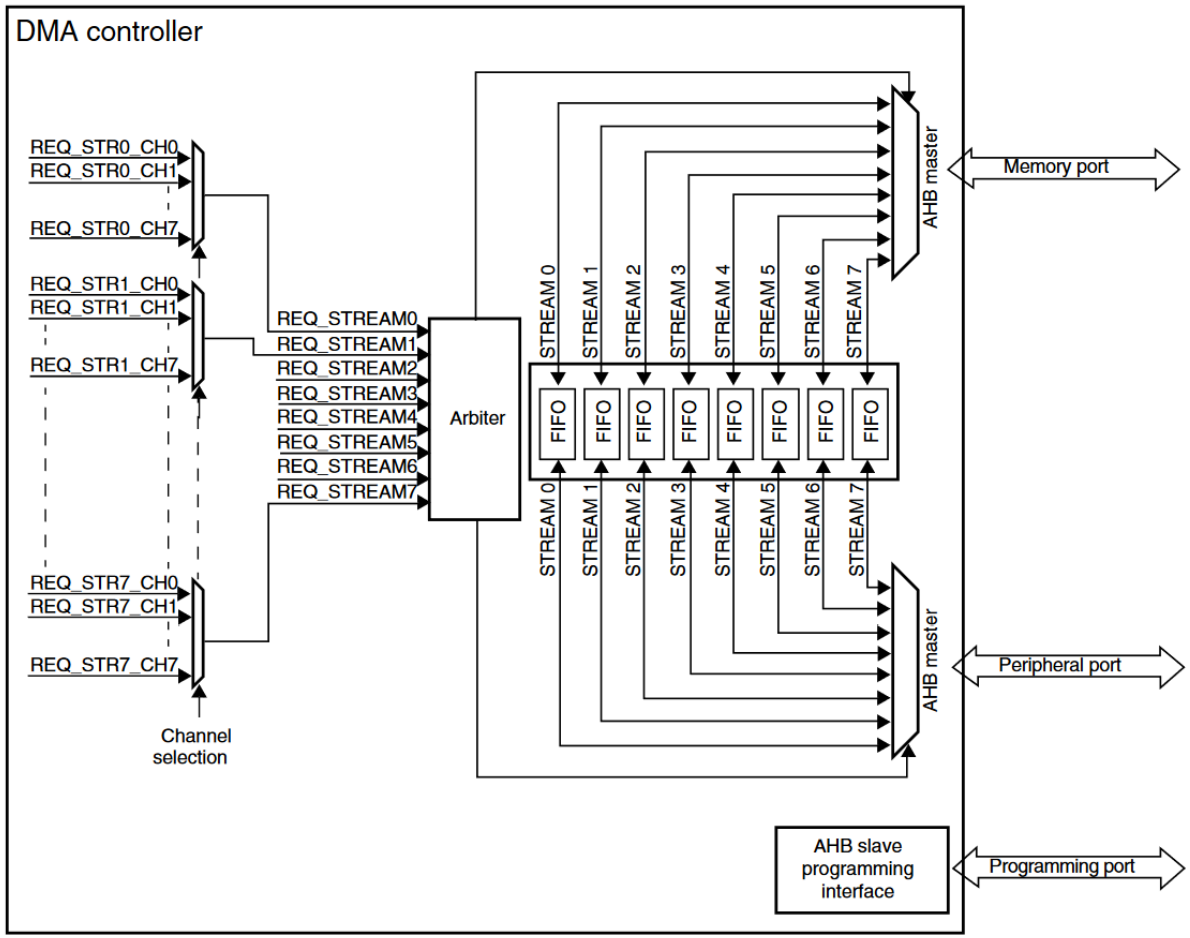
Wrocław University
of Science and Technology

Outline

Wrocław University of Science and Technology

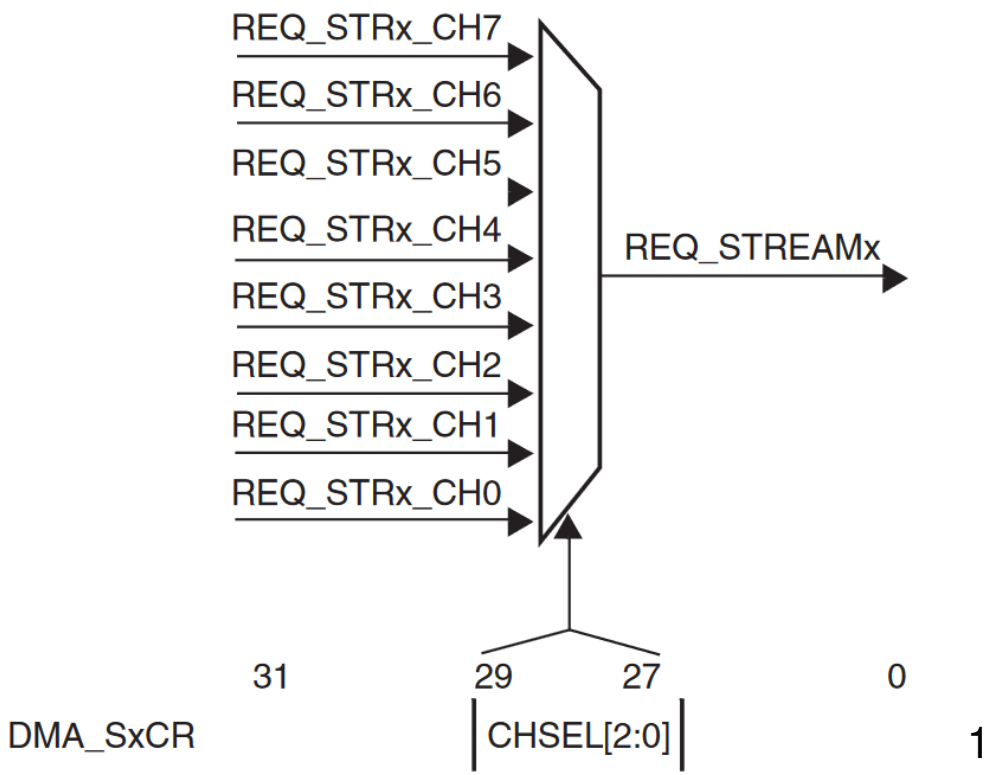


DMA (1/14)


DMA controller
REQ_STR0_CH0
REQ_STR0_CH1
REQ_STR0_CH7
REQ_STR1_CH0
REQ_STR1_CH1
REQ_STR1_CH7
REQ_STR7_CH0
REQ_STR7_CH1
REQ_STR7_CH7
Channel selection
REQ_STREAM0
REQ_STREAM1
REQ_STREAM2
REQ_STREAM3
REQ_STREAM4
REQ_STREAM5
REQ_STREAM6
REQ_STREAM7
Arbiter
STREAM 0
STREAM 1
STREAM 2
STREAM 3
STREAM 4
STREAM 5
STREAM 6
STREAM 7
FIFO
AHB master
Memory port
Peripheral port
AHB slave programming interface
Programming port

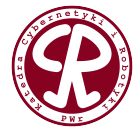
1

DMA (2/14)



W przypadku mikrokontrolerów np. STM32F1, STM32F3, . . . wyłącznie **jeden** kanał na żądanie może być aktywny.

Wrocław University of Science and Technology

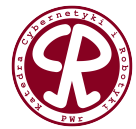


DMA (3/14)

Peripheral requests	Stream 0	Stream 1	Stream 2	Stream 3	Stream 4	Stream 5	Stream 6	Stream 7
Channel 0	SPI3_RX	-	SPI3_RX	SPI2_RX	SPI2_TX	SPI3_TX	-	SPI3_TX
Channel 1	I2C1_RX	-	TIM7_UP		TIM7_UP	I2C1_RX	I2C1_TX	I2C1_TX
Channel 2	TIM4_CH1	-	I2S3_EXT _RX	TIM4_CH2	I2S2_EXT_ TX	I2S3_EXT _TX	TIM4_UP	TIM4_CH3
Channel 3	I2S3_EXT _RX	TIM2_UP TIM2_CH3	I2C3_RX	I2S2_EXT_ RX	I2C3_TX	TIM2_CH1	TIM2_CH2 TIM2_CH4	TIM2_UP TIM2_CH4
Channel 4	UART5 _RX	USART3 _RX	UART4 _RX	USART3 _TX	UART4 _TX	USART2 _RX	USART2 _TX	UART5 _TX
Channel 5	UART8 _TX	UART7 _TX	TIM3_CH4 TIM3_UP	UART7 _RX	TIM3_CH1 TIM3_TRIG	TIM3_CH2	UART8_ RX	TIM3_CH3
Channel 6	TIM5_CH3 TIM5_UP	TIM5_CH4 TIM5_TRIG	TIM5_CH1	TIM5_CH4 TIM5_TRIG	TIM5_CH2	-	TIM5_UP	-
Channel 7	-	TIM6_UP	I2C2_RX	I2C2_RX	USART3_ TX	DAC1	DAC2	I2C2_TX

1

Wrocław University of Science and Technology



DMA (4/14)

Kierunki transmisji danych:

- z peryferium do pamięci,
- z pamięci do peryferium,
- *z pamięci do pamięci*.

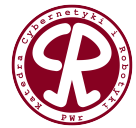
Wrocław University of Science and Technology

DMA (5/14)

Długość danych źródłowych i docelowych:

- Byte (8 bitów, 1 bajt),
- Half-word (16 bitów, 2 bajty),
- Word (32 bity, 4 bajty).

Wrocław University of Science and Technology

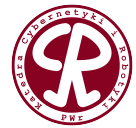


DMA (6/14)

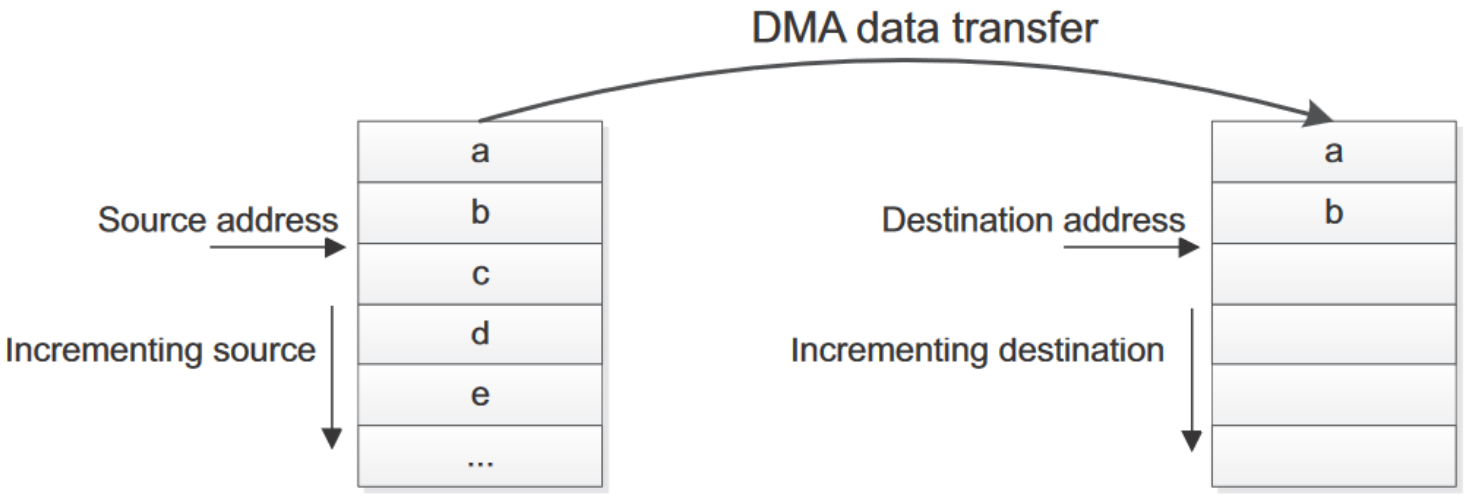
Tryby transferu:

- tryb normalny,
- tryb cykliczny.

Wrocław University of Science and Technology

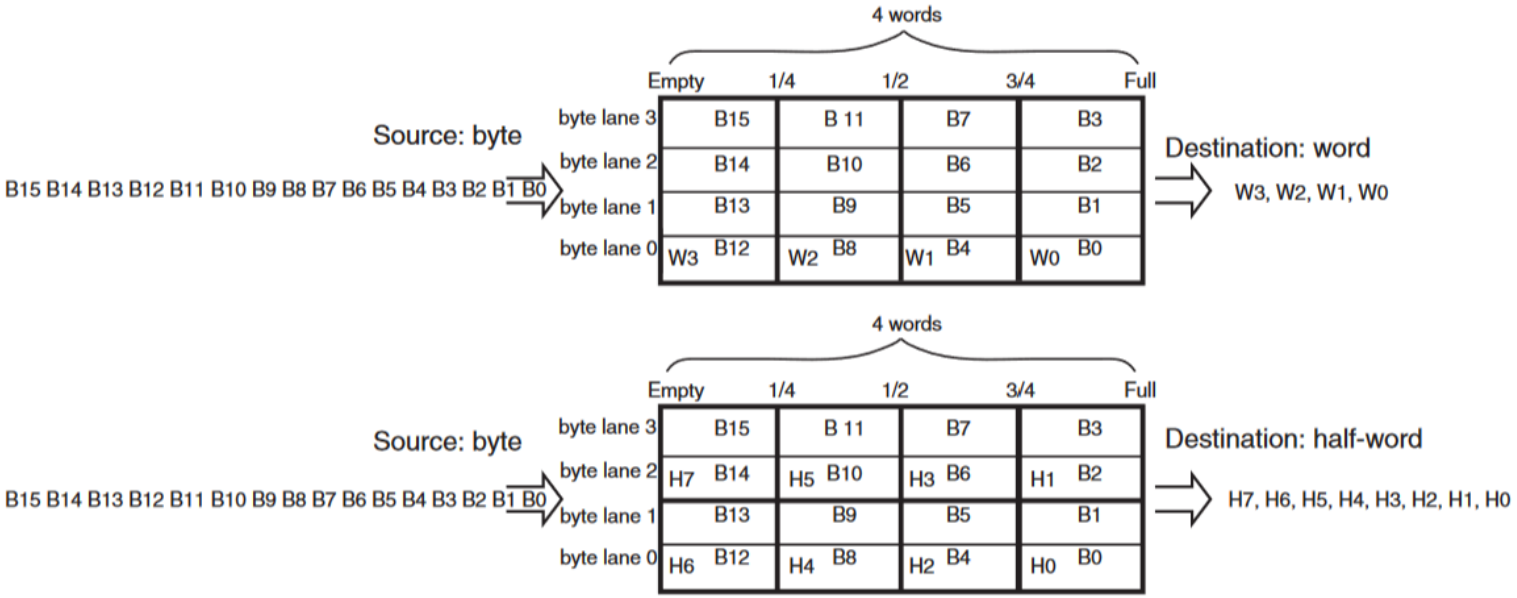


DMA (7/14)


DMA data transfer
Source address
Incrementing source
a
b
c
d
e
...
Destination address
Incrementing destination
a
b

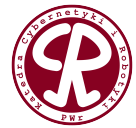
1

DMA (8/14)

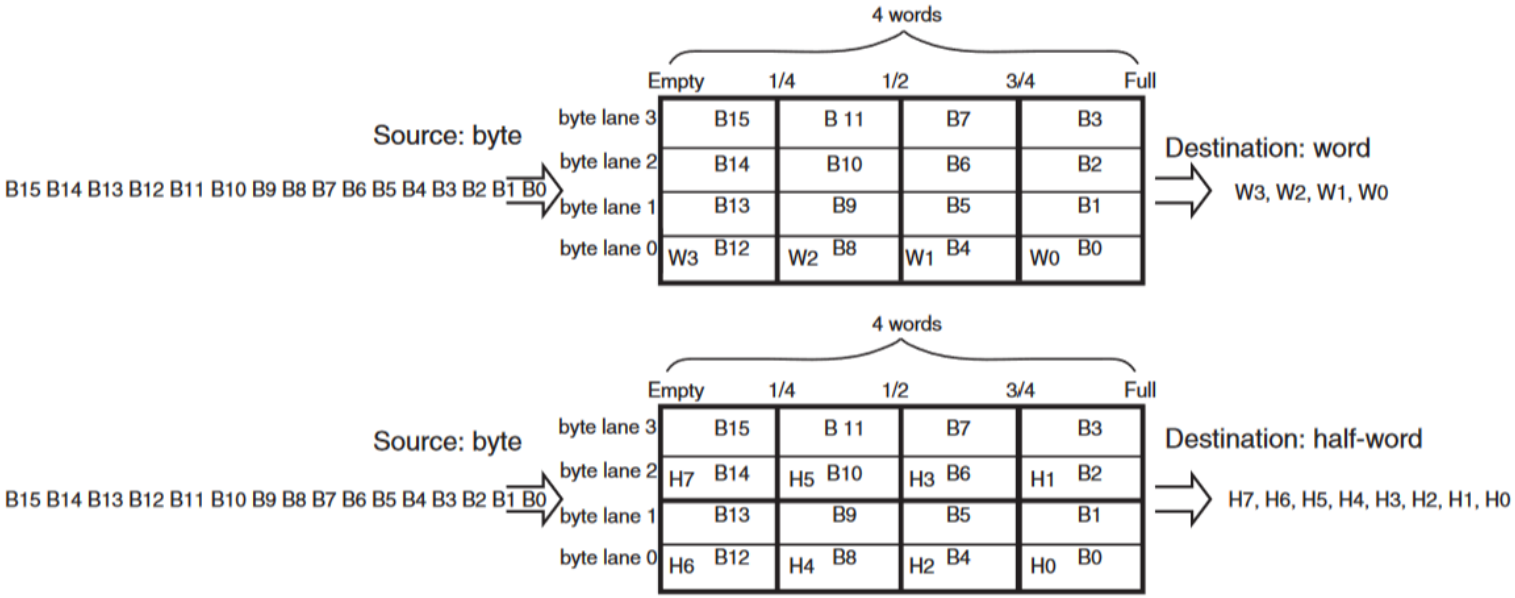


1

Wrocław University
of Science and Technology



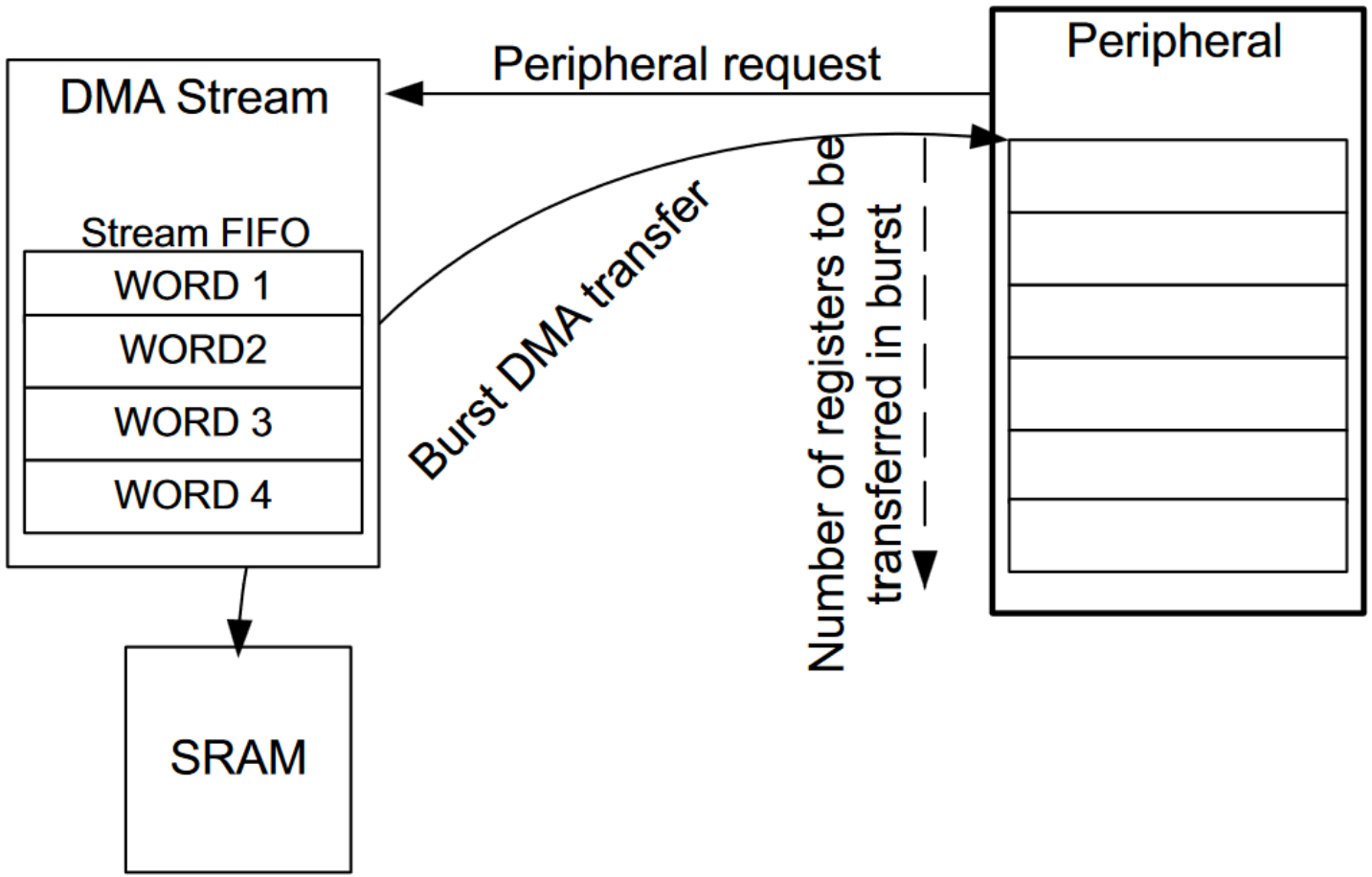
DMA (9/14)


4 words
Empty
1/4
1/2
3/4
Full
Source: byte
B15 B14 B13 B12 B11 B10 B9 B8 B7 B6 B5 B4 B3 B2 B1 B0
byte lane 3
byte lane 2
byte lane 1
byte lane 0
B15 B 11 B7 B3
B14 B10 B6 B2
B13 B9 B5 B1
W3 B12 W2 B8 W1 B4 W0 B0
Destination: word
W3, W2, W1, W0
4 words
Empty
1/4
1/2
3/4
Full
Source: byte
B15 B14 B13 B12 B11 B10 B9 B8 B7 B6 B5 B4 B3 B2 B1 B0
byte lane 3
byte lane 2
byte lane 1
byte lane 0
B15 B 11 B7 B3
H7 B14 H5 B10 H3 B6 H1 B2
B13 B9 B5 B1
H6 B12 H4 B8 H2 B4 H0 B0
Destination: half-word
H7, H6, H5, H4, H3, H2, H1, H0

1

DMA (10/14)

Transfer typu *Burst*.



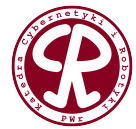
DMA (11/14)

Dozwolone konfiguracje transferu seryjnego (z ang. *burst transfer*)

MSIZE	FIFO level	MBURST = INCR4	MBURST = INCR8	MBURST = INCR16
Byte	1/4	1 burst of 4 bytes	forbidden	forbidden
	1/2	2 bursts of 4 bytes	1 burst of 8 bytes	
	3/4	3 bursts of 4 bytes	forbidden	
	Full	4 bursts of 4 bytes	2 bursts of 8 bytes	1 burst of 16 bytes
Half-word	1/4	forbidden	forbidden	forbidden
	1/2	1 burst of 4 half-words		
	3/4	forbidden		
	Full	2 bursts of 4 half-words	1 burst of 8 Half-word	
Word	1/4	forbidden	forbidden	
	1/2			
	3/4			
	Full	1 burst of 4 words		

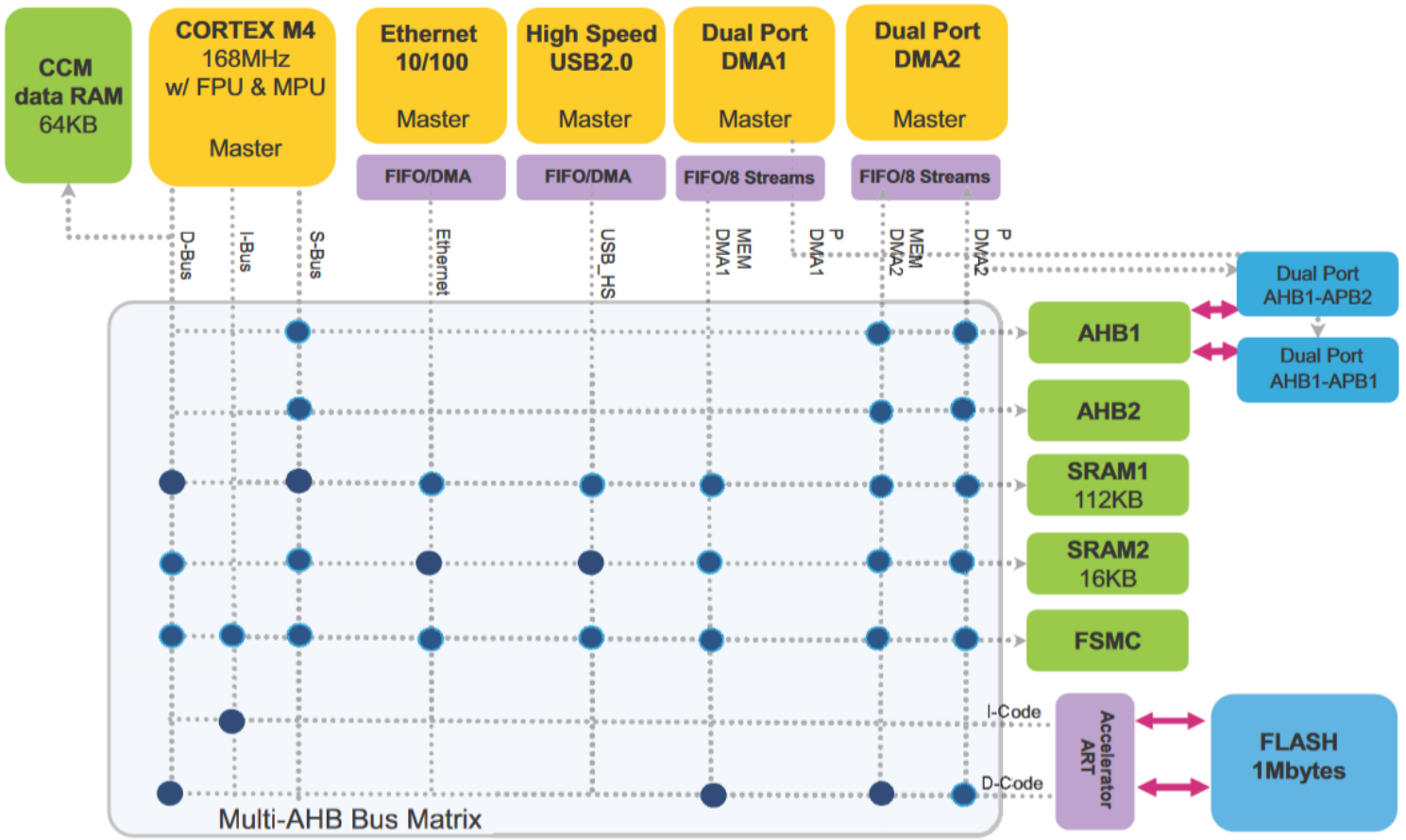
Wrocław University of Science and Technology

1



DMA (12/14)

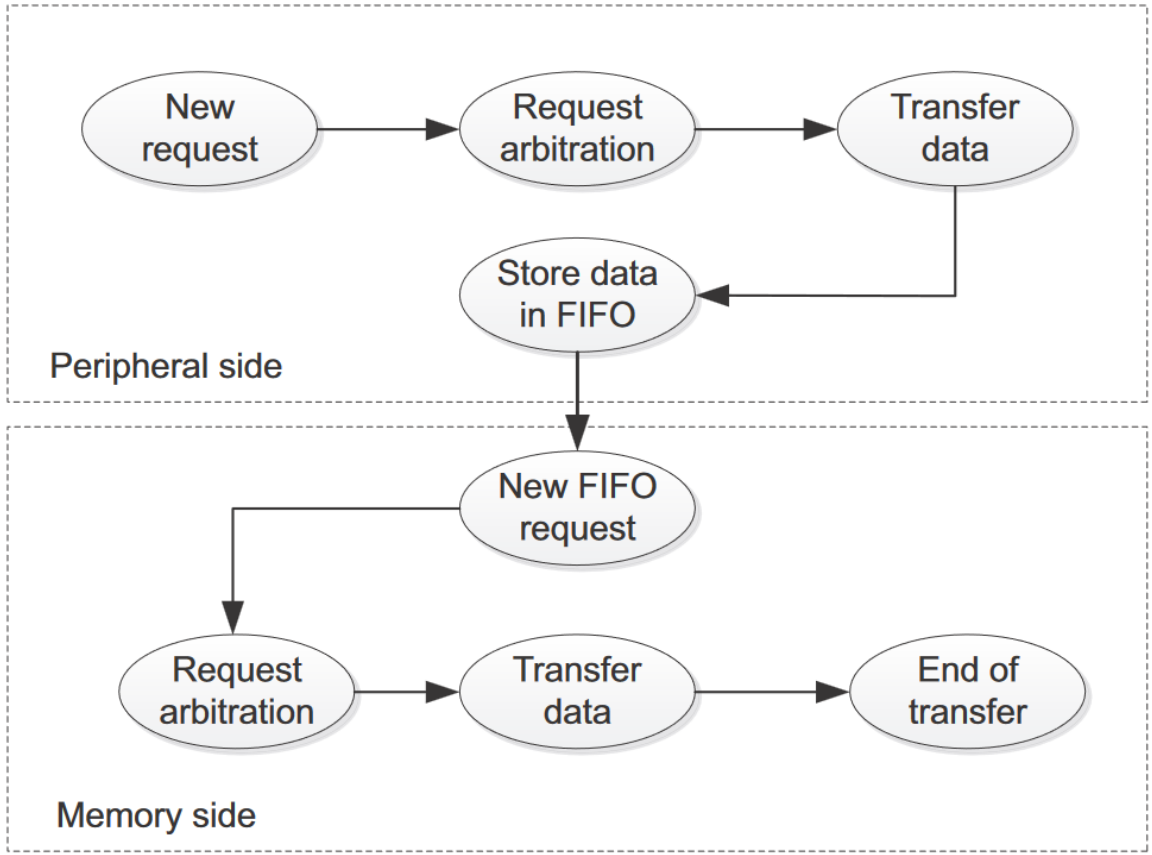
Architektura systemu połączeń



1

DMA (13/14)

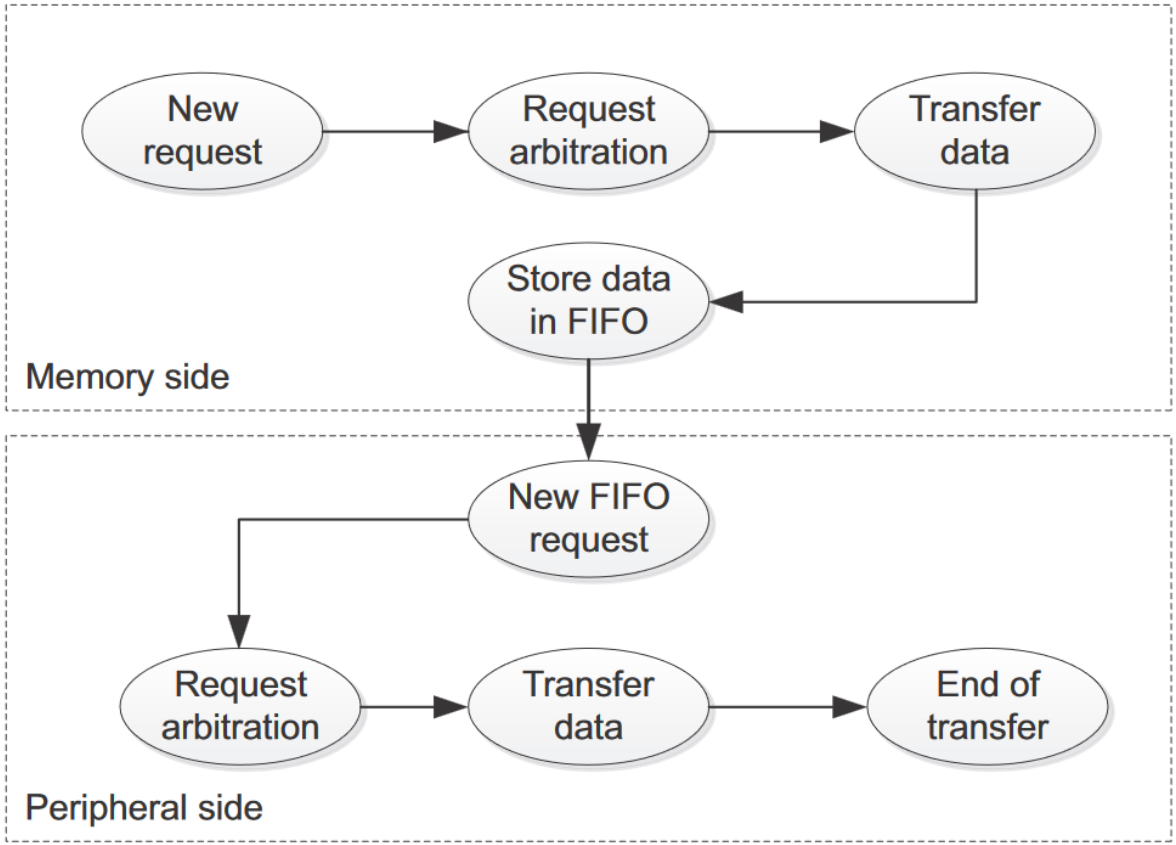
Maszyna stanów w przypadku transferu z peryferium do pamięci



1

DMA (14/14)

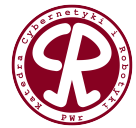
Maszyna stanów w przypadku transferu z pamięci do pamięci



[1]

Wrocław University of Science and Technology

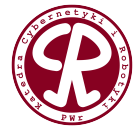
[1] http://www.st.com



API (1/1)

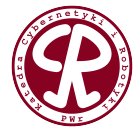
```
1  HAL_DMA_Start
2  HAL_DMA_Start_IT
3  HAL_DMA_Abort
4  HAL_DMA_Abort_IT
5  HAL_DMA_PollForTransfer
6  HAL_DMA_IRQHandler
7  HAL_DMA_RegisterCallback
8  HAL_DMA_UnRegisterCallback
9
10 __HAL_DMA_ENABLE()
11 __HAL_DMA_DISABLE();
12 __HAL_DMA_ENABLE_IT()
13 __HAL_DMA_DISABLE_IT();
14 __HAL_DMA_GET_IT_SOURCE();
15 __HAL_DMA_GET_COUNTER();
```

Wrocław University of Science and Technology

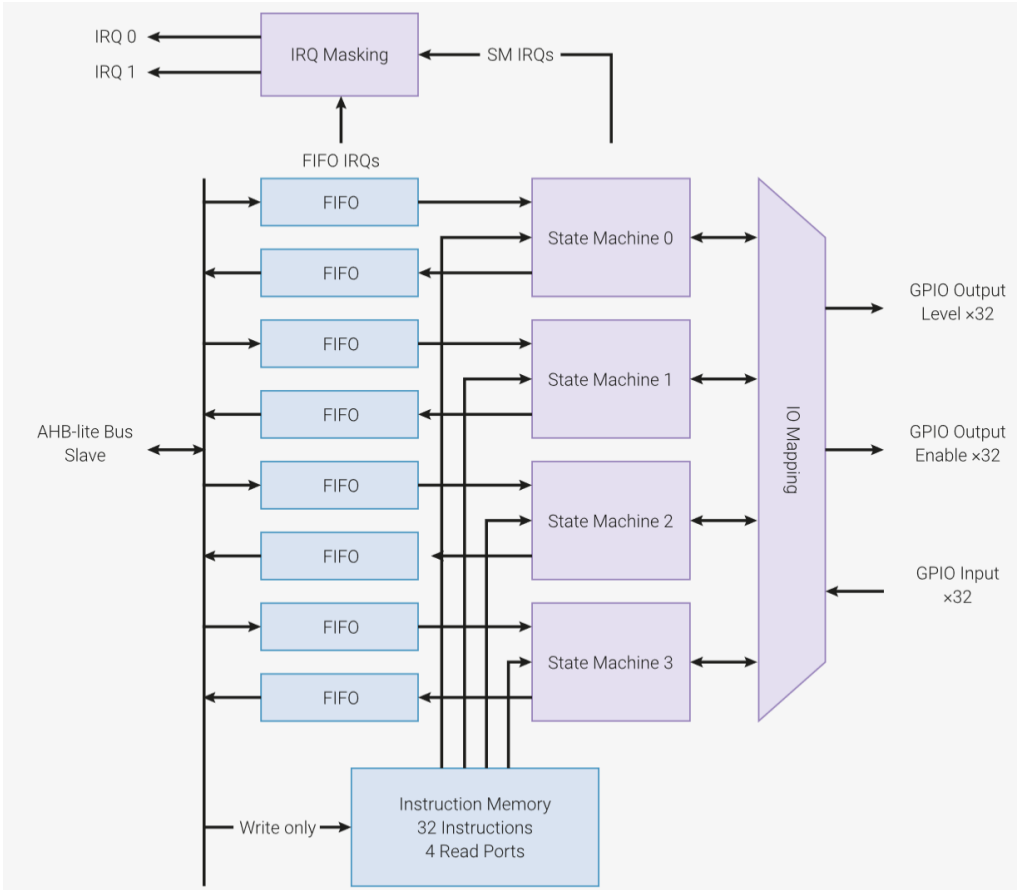


Outline

Wrocław University of Science and Technology



PIO

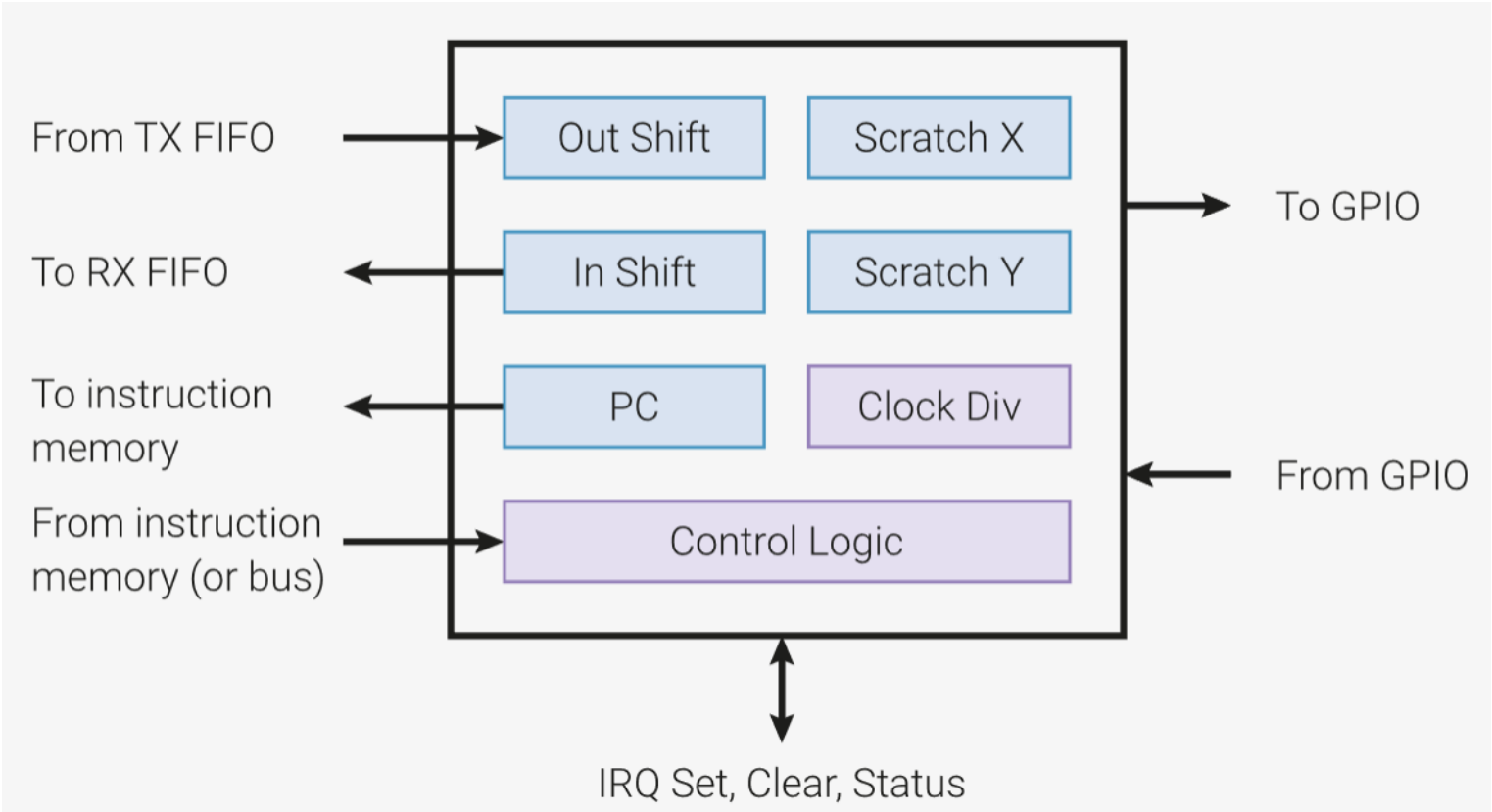


5

Wrocław University of Science and Technology

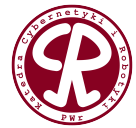
[5] https://raspberrypi.com

Maszyna stanów



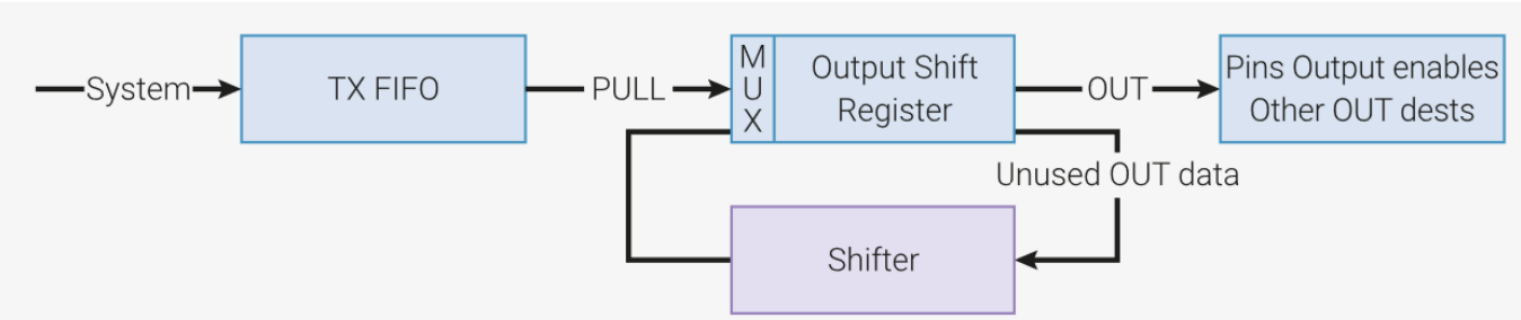
[6]

Wrocław University of Science and Technology



[6] https://raspberrypi.com

Output Shift Register (OSR) (1/2)



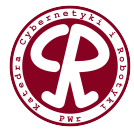
7

Output Shift Register (OSR) (2/2)

Rejestr Przesunięcia Wyjściowego (OSR) przechowuje i przesuwa dane wyjściowe między TX FIFO a pinami (lub innymi miejscami docelowymi, takimi jak rejestry tymczasowe).

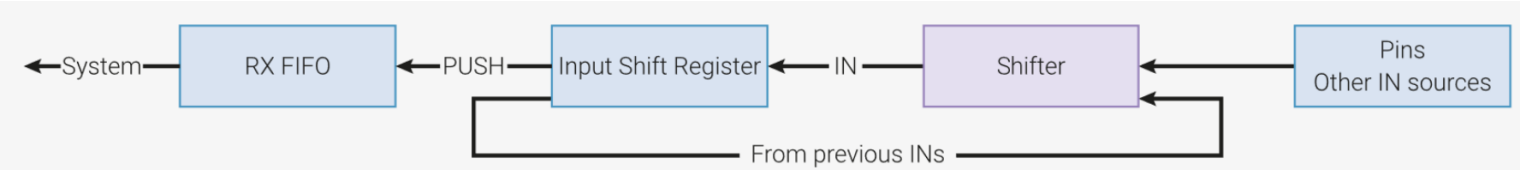
- Instrukcje PULL: usuń 32-bitowe słowo z TX FIFO i umieść je w OSR.
- Instrukcje OUT: przesuwają dane z OSR do innych miejsc docelowych, 1...32 bity na raz.
- OSR wypełnia się zerami podczas przesuwania danych na zewnątrz.
- Maszyna stanowa automatycznie uzupełni OSR z FIFO na instrukcji OUT, gdy osiągnięty zostanie pewien próg łącznej liczby przesunięć, jeśli autopull jest włączony.
- Kierunek przesunięcia może być w lewo/w prawo, konfigurowany przez procesor za pomocą rejestrów konfiguracyjnych.

Wrocław University of Science and Technology



[7] https://raspberrypi.com

Input Shift Register (ISR) (1/3)



8

Input Shift Register (ISR) (2/3)

- Instrukcje IN przesuwają 1...32 bity na raz do rejestru.
- Instrukcje PUSH zapisują zawartość ISR do RX FIFO.
- ISR jest wypełniany zerami podczas operacji PUSH.
- Maszyna stanów automatycznie wykonuje operację PUSH na ISR przy instrukcji IN, gdy osiągnięty zostanie pewien próg przesunięć, jeśli autopush jest włączony.
- Kierunek przesunięcia jest konfigurowalny przez procesor za pomocą rejestrów konfiguracyjnych.

Niektóre urządzenia peryferyjne, takie jak UART, muszą przesuwać dane z lewej strony, aby uzyskać poprawny porządek bitów, ponieważ kolejność przewodów jest najpierw od najmniej znaczącego bitu (LSB-first); jednak procesor może oczekiwać, że wynikowy bajt będzie wyrównany do prawej.

Wrocław University of Science and Technology

Input Shift Register (ISR) (3/3)

Problem ten jest rozwiązywany dzięki specjalnemu źródle danych wejściowych NULL. Pozwala ono przesłać pewną liczbę zer do ISR po przesunięciu danych.

Wrocław University
of Science and Technology

[8] https://raspberrypi.com

Instrukcje PIO

Dostępne instrukcje w maszynie stanów PIO

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JMP	0	0	0	Delay/side-set					Condition			Address				
WAIT	0	0	1	Delay/side-set					Pol	Source		Index				
IN	0	1	0	Delay/side-set					Source			Bit count				
OUT	0	1	1	Delay/side-set					Destination			Bit count				
PUSH	1	0	0	Delay/side-set					0	IfF	Blk	0	0	0	0	0
PULL	1	0	0	Delay/side-set					1	IfE	Blk	0	0	0	0	0
MOV	1	0	1	Delay/side-set					Destination			Op		Source		
IRQ	1	1	0	Delay/side-set					0	Clr	Wait	Index				
SET	1	1	1	Delay/side-set					Destination			Data				

[9]

Wrocław University of Science and Technology

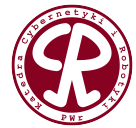
[9] https://raspberrypi.com

Instrukcja skoku JMP (1/2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JMP	0	0	0	Delay/side-set					Condition			Address				

10

Wrocław University
of Science and Technology



Instrukcja skoku JMP (2/2)

Ustaw licznik programu na adres, jeśli warunek jest prawdziwy, w przeciwnym razie brak operacji. Opóźnienia cyklu w instrukcji JMP zawsze mają miejsce, bez względu na to, czy warunek jest prawdziwy, czy fałszywy, i występują po ocenie warunku oraz zaktualizowaniu licznika programu.

Warunek :

- 000 : (brak warunku): Zawsze
- 001 : !X: Rejestr X równy zeru
- 010 : X–: Rejestr X różny od zera, przed dekrementacją
- 011 : !Y: Rejestr Y równy zeru
- 100 : Y–: Rejestr Y różny od zera, przed dekrementacją
- 101 : X!=Y: Rejestr X różny od Rejestru Y
- 110 : PIN: Skok na pinie wejściowym
- 111 : !OSRE: Wyjściowy rejestr przesunięć nie jest pusty

Adres : Adres instrukcji, do której należy skoczyć. W kodowaniu instrukcji jest to absolutny adres w pamięci instrukcji PIO.

Wrocław University of Science and Technology

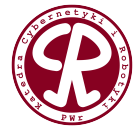
[10] https://raspberrypi.com

Instrukcja opóźnienia WAIT (1/4)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WAIT	0	0	1	Delay/side-set					Pol	Source		Index				

11

Wrocław University
of Science and Technology



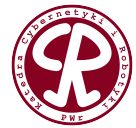
Instrukcja opóźnienia WAIT (2/4)

Czeka, aż zostanie spełniony pewien warunek. Tak jak wszystkie instrukcje opóźnienia, cykle opóźnienia rozpoczynają się po zakończeniu instrukcji. Oznacza to, że jeśli występują jakiekolwiek cykle opóźnienia, nie zaczynają one odliczania, dopóki warunek oczekiwania nie zostanie spełniony.

Polarity :

- 1 : Oczekiwanie na 1.
- 0 : Oczekiwanie na 0.

Wrocław University of Science and Technology



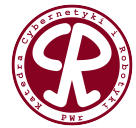
Instrukcja opóźnienia WAIT (3/4)

Source : na co oczekiwać.

- 00 : GPIO: Wybór wejścia GPIO systemu przez Index. Bezwzględny indeks GPIO i nie jest objęty odwzorowaniem wejść maszyny stanu.
- 01 : PIN: Wejściowy pin wybrany przez Index. To odwzorowanie wejść maszyny stanu jest stosowane najpierw, a następnie Index wybiera, na których z odwzorowanych bitów oczekiwać. Innymi słowy, pin jest wybierany przez dodanie Indeksu do konfiguracji PINCTRL_IN_BASE, modulo 32.
- 10 : IRQ: Flaga PIO IRQ wybrana przez Index
- 11 : Zarezerwowane

Index : który pin lub bit sprawdzić

Wrocław University of Science and Technology



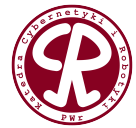
Instrukcja opóźnienia WAIT (4/4)

WAIT x IRQ zachowuje się nieco inaczej niż inne źródła WAIT:

- Jeśli polaryzacja wynosi 1, wybrana flaga IRQ jest kasowana przez maszynę stanu po spełnieniu warunku oczekiwania.
- Indeks flagi jest dekodowany w taki sam sposób jak pole indeksu IRQ: jeśli MSB jest ustawione, ID maszyny stanu (0...3) jest dodawane do indeksu IRQ, za pomocą dodawania modulo-4 na dwóch bitach LSB. Na przykład maszyna stanów 2 z wartością flagi '0x11' będzie oczekiwać na flagę 3, a wartość flagi '0x13' będzie oczekiwać na flagę 1. Pozwala to wielu maszynom stanu działającym tym samym programem na synchronizację między sobą.

Wrocław University of Science and Technology

[11] https://raspberrypi.com

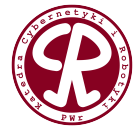


Instrukcja odczytu IN (1/2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IN	0	1	0	Delay/side-set					Source			Bit count				

12

Wrocław University
of Science and Technology



Instrukcja odczytu IN (2/2)

Przesuń *Bit count* bitów ze źródła do ISR. Kierunek przesunięcia jest konfigurowany dla każdej maszyny stanu przez SHIFTCTRL_IN_SHIFTDIR. Dodatkowo zwiększ liczbę przesunięć wejściowych o *Bit count*, nasycając do 32.

Source :

- 000 : PINS
- 001 : X (rejestr tymczasowy X)
- 010 : Y (rejestr tymczasowy Y)
- 011 : NULL (zera)
- 100 : Zarezerwowane
- 101 : Zarezerwowane
- 110 : ISR
- 111 : OSR

Bit count : Ile bitów przesunąć do ISR. 1...32 bity, 32 jest zakodowane jako 00000

Wrocław University of Science and Technology

[12] https://raspberrypi.com

Instrukcja zapisu OUT (1/2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OUT	0	1	1	Delay/side-set					Destination			Bit count				

13

Wrocław University
of Science and Technology

Instrukcja zapisu OUT (2/2)

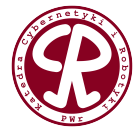
Przesuń *Bit count* bitów z Rejestru Przesunięcia Wyjściowego (OSR) i zapisz te bity do Destynacji. Dodatkowo zwiększ liczbę przesunięć wyjściowych o *Bit count*, nasycając do 32.

estination :

- 000 : PINS
- 001 : X (rejestr tymczasowy X)
- 010 : Y (rejestr tymczasowy Y)
- 011 : NULL (porzucenie danych)
- 100 : PINDIRS
- 101 : PC
- 110 : ISR (ustawia także licznik przesunięć ISR na wartość *Bit count*)
- 111 : EXEC (Wykonaj dane przesunięcia OSR jako instrukcję)

Bit count : ile bitów przesunąć z OSR. 1...32 bity, 32 jest zakodowane jako 00000.

Wrocław University of Science and Technology



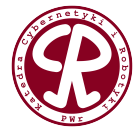
[13] https://raspberrypi.com

Instrukcja wysyłki PUSH (1/2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUSH	1	0	0	Delay/side-set					0	IfF	Blk	0	0	0	0	0

14

Wrocław University of Science and Technology



Instrukcja wysyłki PUSH (2/2)

Wprowadź zawartość ISR do RX FIFO jako pojedyncze 32-bitowe słowo. Wyczyść ISR ustawiając zera.

IfFull : Jeśli 1, nie rób nic, chyba że całkowita liczba przesunięć wejściowych osiągnie próg SHIFTCTRL_PUSH_THRESH (to samo co dla autopush).

Block : Jeśli 1, zatrzymaj wykonanie, jeśli RX FIFO jest pełne.

Wrocław University of Science and Technology

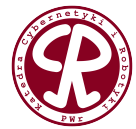
[14] https://raspberrypi.com

Instrukcja odbioru PULL (1/2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PULL	1	0	0	Delay/side-set					1	IfE	Blk	0	0	0	0	0

15

Wrocław University of Science and Technology



Instrukcja odbioru PULL (2/2)

Wczytaj 32-bitowe słowo z TX FIFO do OSR.

IfEmpty : Jeśli 1, nie rób nic, chyba że całkowita liczba przesunięć wyjściowych osiągnie próg SHIFTCTRL_PULL_THRESH (to samo co dla autopull).

Block : Jeśli 1, zatrzymaj się, jeśli TX FIFO jest puste. Jeśli 0, odczyt z pustego FIFO kopiuje rejestr tymczasowy X do OSR.

Wrocław University of Science and Technology

[15] https://raspberrypi.com

Instrukcja kopii MOV (1/3)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOV	1	0	1	Delay/side-set					Destination			Op		Source		

16

Wrocław University
of Science and Technology

Instrukcja kopii MOV (2/3)

estination :

- 000 : PINS (Używa tego samego odwzorowania pinów co OUT)
- 001 : X (Rejestr tymczasowy X)
- 010 : Y (Rejestr tymczasowy Y)
- 011 : Zarezerwowane
- 100 : EXEC (Wykonaj dane jako instrukcję)
- 101 : PC
- 110 : ISR (Licznik jest resetowany do 0 przez tę operację; pusty)
- 111 : OSR (Licznik jest resetowany do 0 przez tę operację; pełny)

Op :

- 00 : Brak
- 01 : Inwersja (bitowe dopełnienie)
- 10 : Odwróć bity
- 11 : Zarezerwowane

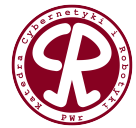
Wrocław University of Science and Technology

Instrukcja kopii MOV (3/3)

Source :

- 000 : PINS (Używa tego samego odwzorowania pinów co IN)
- 001 : X
- 010 : Y
- 011 : NULL
- 100 : Zarezerwowane
- 101 : STATUS
- 110 : ISR
- 111 : OSR

Wrocław University of Science and Technology



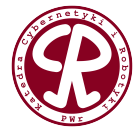
[16] https://raspberrypi.com

Instrukcja przerwania IRQ (1/2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRQ	1	1	0	Delay/side-set					0	Clr	Wait	Index				

17

Wrocław University
of Science and Technology



Instrukcja przerwania IRQ (2/2)

Ustaw lub wyczyść flagę IRQ wybraną przez argument *Index*.

Clear : jeśli 1, wyczyść flagę wybraną przez *Index*, zamiast ją podnosić. Jeśli *Clear* jest ustawione, bit *Wait* nie ma efektu.

Wait : jeśli 1, zatrzymaj się, aż podniesiona flaga zostanie ponownie obniżona, na przykład jeśli system przerwań potwierdzi flagę.

Index :

- Trzy najmniej znaczące bity określają indeks IRQ od 0 do 7. Ta flaga IRQ będzie ustawiona/wyczyszczona w zależności od bitu *Clear*.
- Jeśli MSB jest ustawione, ID maszyny stanu (0...3) jest dodawane do indeksu IRQ, za pomocą dodawania modulo-4 na dwóch LSB. Na przykład maszyna stanu 2 z wartością flagi 0x11 podniesie flagę 3, a wartość flagi 0x13 ustawi flagę 1.

Wrocław University of Science and Technology

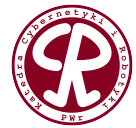
[17] https://raspberrypi.com

Instrukcja wpisu SET (1/2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SET	1	1	1	Delay/side-set					Destination			Data				

18

Wrocław University
of Science and Technology



Instrukcja wpisu SET (2/2)

Zapisz natychmiastową wartość *Data* do *Destination*.

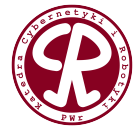
estination :

- 00 : PINS
- 01 : X (rejestr tymczasowy X) 5 bitów LSB jest ustawionych na wartość *Data*, pozostałe są zerowane.
- 10 : Y (rejestr tymczasowy Y) 5 bitów LSB jest ustawionych na wartość *Data*, pozostałe są zerowane.
- 11 : Zarezerwowane

Data : 5-bitowa wartość do przekazania do pinów lub rejestru.

Wrocław University of Science and Technology

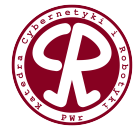
[18] https://raspberrypi.com



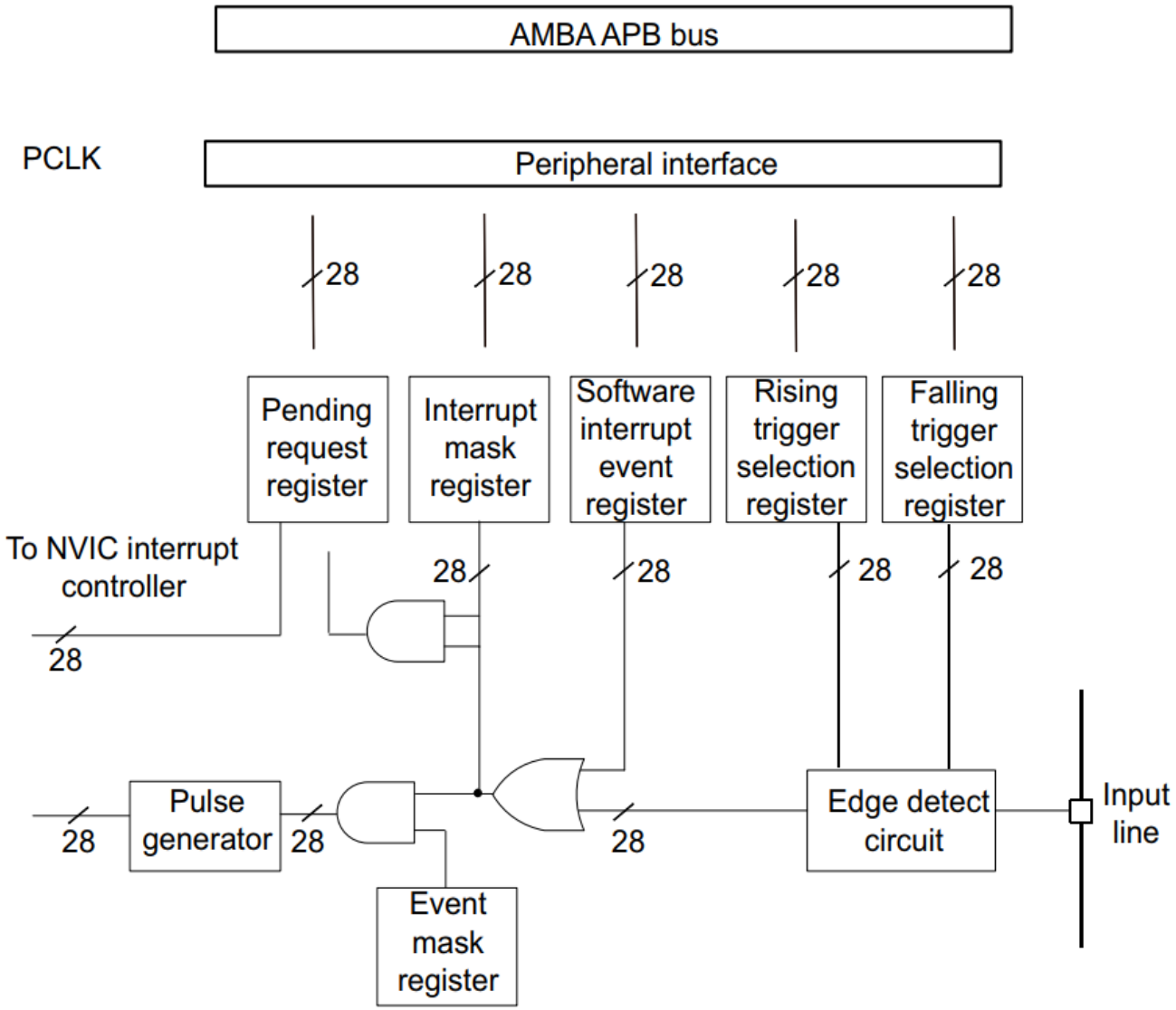
EXTI

- Obsługa generowania do 36 żądań zdarzeń/przerwań
- Niezależna konfiguracja każdej linii jako zewnętrzne lub wewnętrzne żądanie zdarzenia
- Niezależna maska dla każdej linii zdarzenia/przerwania
- Automatyczne wyłączanie wewnętrznych linii, gdy system nie jest w trybie STOP
- Niezależny wyzwalacz dla zewnętrznej linii zdarzenia/przerwania
- Dedykowany bit statusu dla zewnętrznej linii przerwania
- Emulacja dla wszystkich zewnętrznych żądań zdarzeń

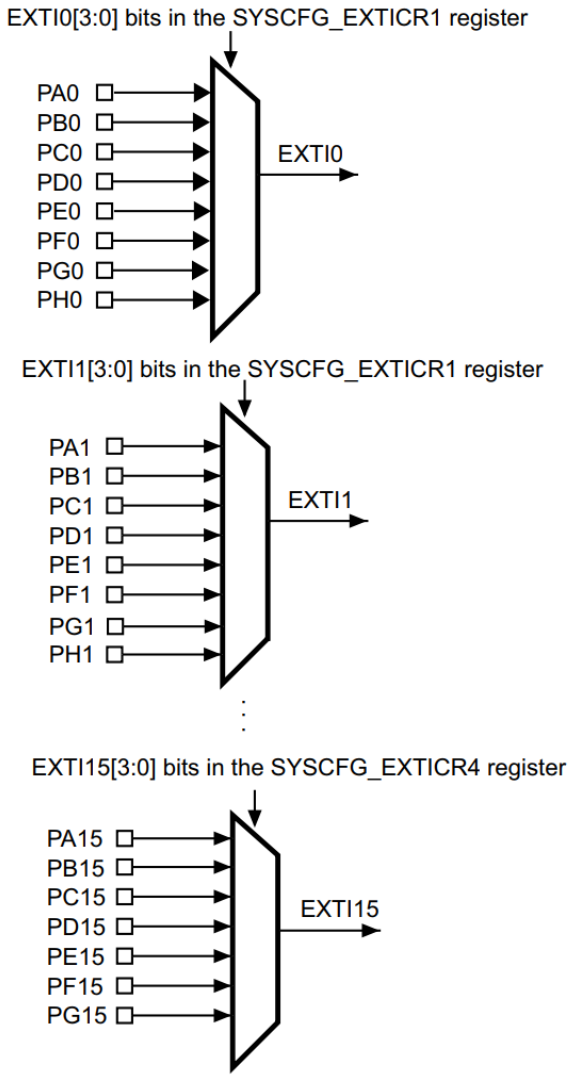
Wrocław University of Science and Technology



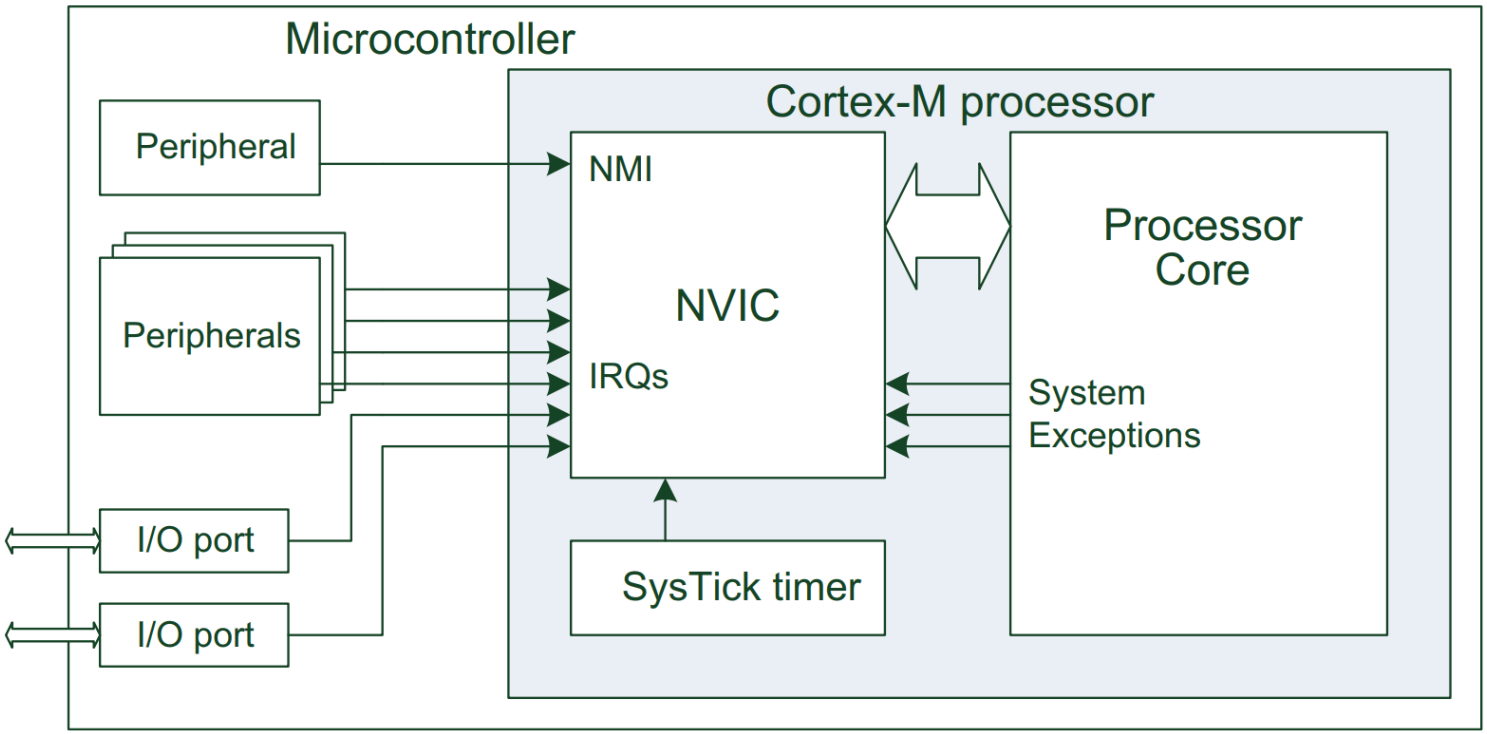
Połączenia wewnętrzne



Linie przerwań

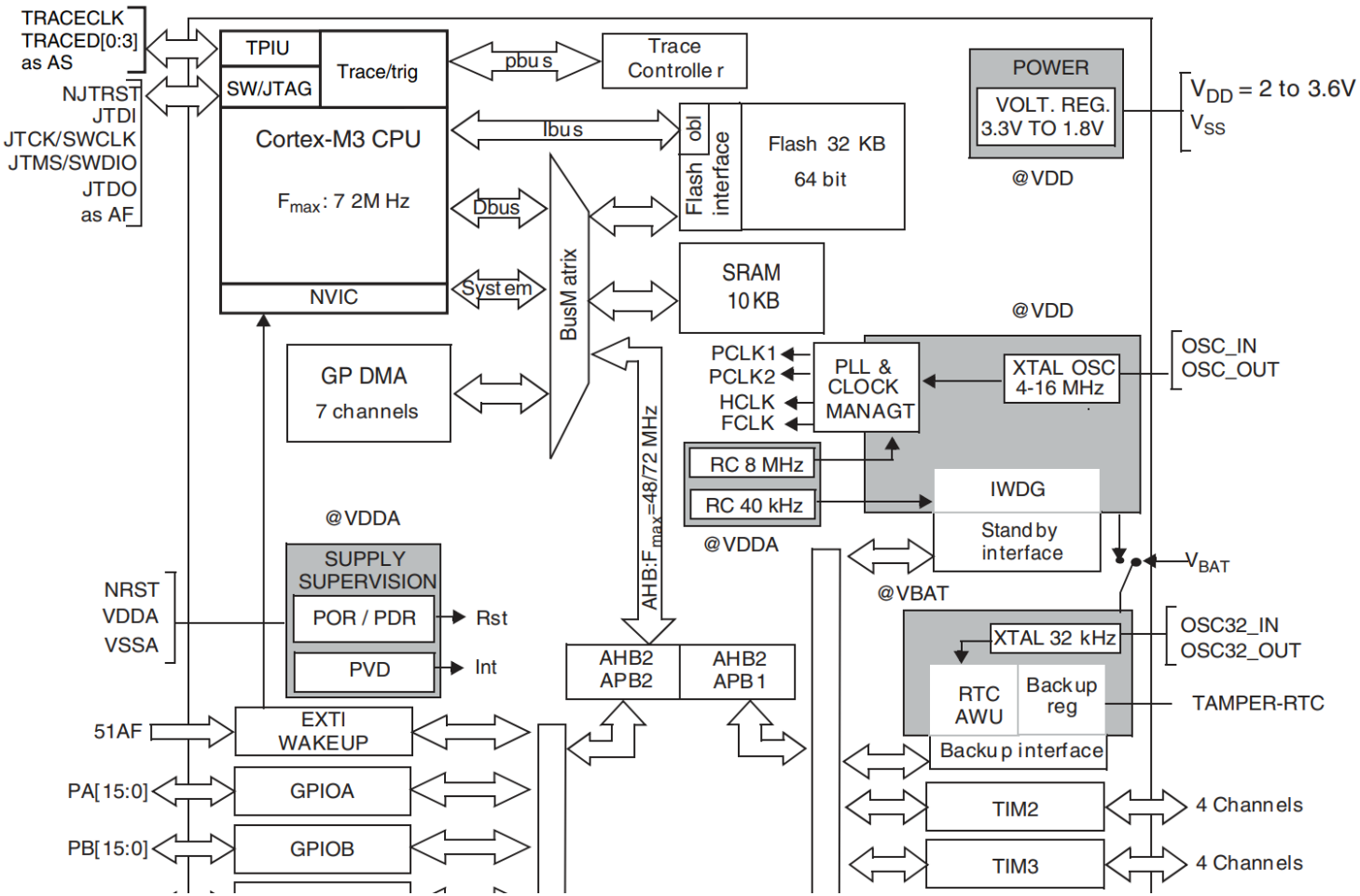


NVIC (1/2)


Microcontroller
Cortex-M processor
Peripheral
Peripherals
I/O port
I/O port
NMI
NVIC
IRQs
SysTick timer
Processor Core
System Exceptions

21

NVIC (2/2)



22

Wrocław University of Science and Technology

21

J. Yiu, The definitive guide to ARM Cortex-M3 and Cortex-M4 processors

22 st.com

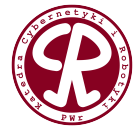
Przerwania systemowe (1/4)

1. Reset (-3), najwyższy priorytet, reset układu,
2. NMI (-2), Non-Maskable Interrupt, generowane przez peryferia, bądź pochodzą z zewnętrznego źródła,
3. Hard Fault, -1, Obsługa wszystkich błędów,

- od tego momentu rozpoczynają się przerwania, których priorytet jest programowalny,

4. MemManage Fault, błąd zarządzania pamięcią; naruszenie MPU lub wykonanie programu z lokalizacji adresowych o atrybucie pamięci XN (eXecute Never),

Wrocław University of Science and Technology



Przerwania systemowe (2/4)

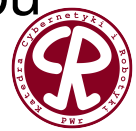
5 Bus Fault, błąd magistrali; zazwyczaj występuje, gdy interfejs AHB otrzymuje odpowiedź błędu od urządzenia podrzędnego magistrali (zwany również przerwaniem prefetch abort, jeśli jest to pobranie instrukcji lub przerwaniem danych abort, jeśli jest to dostęp do danych). Może być również spowodowany przez inne „nielegalne” dostępy,

6 Usage Fault, wyjątki związane z błędem programu lub próbą dostępu do koprocesora (procesory Cortex-M3 i Cortex-M4 nie obsługują koprocesora),

7-10 Zarezerwowane,

11 SVC, SuperVisor Call; zwykle używane w środowisku systemu operacyjnego do umożliwienia zadaniom dostępu do usług systemowych,

Wrocław University of Science and Technology



Przerwania systemowe (3/4)

12 Debug Monitor, monitor debugowania; wyjątek dla zdarzeń debugowania, takich jak pułapki (z ang. *breakpoints*), punkty obserwacyjne (z ang. *watchpoints*), gdy stosowane jest programowe rozwiązanie debugowania,

13 Zarezerwowane,

14 PendSV, wywołanie usługi zawieszenia; wyjątek zwykle używany przez system operacyjny w procesach takich jak przełączanie kontekstu,

15 SYSTICK, licznik System Tick; wyjątek generowany przez moduł licznika zegara, który jest zawarty w procesorze. Może być używany przez system operacyjny lub jako prosty moduł licznika,

16 Przerwanie #0, może być generowane przez peryferia w układzie lub z zewnętrznych źródeł,

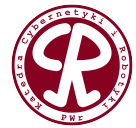
Przerwania systemowe (4/4)

17 Przerwanie #1,

18-254 ...

255 Przerwanie #239.

Wrocław University of Science and Technology



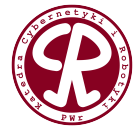
Grupy priorytetów (1/2)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Group priority					Subpriority		

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Implemented			NOT implemented				

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Group priority		Subpriority				NOT implemented	

Wrocław University of Science and Technology



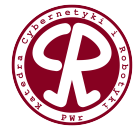
Grupy priorytetów (2/2)

Group priority, dawniej preemption priority, określa czy przerwanie ma być obsłużone jeżeli już inne przerwanie jest obsługiwane. Porównywane są wówczas priorytety grupy. W przypadku, gdy priorytet grupy jest identyczny to pod uwagę brany jest podpriorytet.

Uwaga!

Niższa wartość liczby priorytetu oznacza wyższy priorytet.

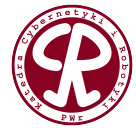
Wrocław University of Science and Technology



Tablica przerwań

Kiedy procesor akceptuje żądanie wyjątku, musi on określić adres początkowy obsługi wyjątku (lub ISR, jeśli wyjątek jest przerwaniem). Informacje te są przechowywane w **tablicy wektorów przerwań** w pamięci.
Domyślnie tabela wektorów zaczyna się od adresu pamięci 0x00000000, a adres wektora jest ułożony zgodnie z numerem wyjątku (pomnożonym przez cztery bajty, 32-bity). Tabela wektorów zazwyczaj jest definiowana w skryptach startowych dostarczanych przez dostawców mikrokontrolerów.

Wrocław University
of Science and Technology

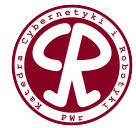


Maskowanie przerwań (1/3)

PRIMASK

W wielu aplikacjach może zajść konieczność tymczasowego wyłączenia wszystkich przerwań w celu wykonania zadań krytycznych pod względem czasu.
Rejestr PRIMASK jest dostępny tylko w trybie uprzywilejowanym.
Wyłącza on wszystkie wyjątki z wykluczeniem NMI i HardFault.
W efekcie zmienia on bieżący poziom priorytetu na 0 (najwyższy poziom programowalny).

Wrocław University of Science and Technology



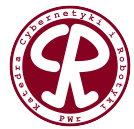
Maskowanie przerwań (2/3)

FAULTMASK

Podobny do PRIMASK, lecz zmienia efektywny bieżący poziom priorytetu na -1, co oznacza, że nawet obsługa HardFault jest zablokowana. ZAtem tylko wyjątek NMI może być wywoływany. Przeznaczony jest do obsługi konfigurowalnych przerwań (tj. MemManage, Bus Fault, Usage Fault), aby podnieść ich priorytet do -1, dzięki czemu mogą mieć one dostęp do niektórych specjalnych funkcji dla wyjątków HardFault, w tym: Pomijanie MPU, Ignorowanie błędów magistrali danych podczas próbkowania urządzenia/pamięci

Podnosząc bieżący poziom priorytetu do -1, FAULTMASK pozwala konfigurowalnym przerwaniom błędów na blokowanie wykonania innych wyjątków podczas obsługi błędów.

Wrocław University of Science and Technology



Maskowanie przerwań (3/3)

BASEPRI

W niektórych przypadkach może być pożądane wyłączenie tylko przerwań o priorytecie niższym niż określony poziom. W takim przypadku należy skorzystać z rejestru BASEPRI. Aby to zrobić, należy zapisać wymagany poziom priorytetu maskującego do rejestru BASEPRI.

Wrocław University of Science and Technology