

ColdFire® Embedded Controllers

MCF528x Family

Architektura mikroprocesorów z rdzeniem
ColdFire



Obsługa sytuacji wyjątkowych (Exception Processing)

Wyjątki

Wyjątek (ang. exception) – mechanizm kontroli przepływu danych występujący w mikroprocesorach oraz we współczesnych językach programowania służący do obsługi zdarzeń wyjątkowych, a w szczególności sytuacji błędnych.

Wyjątki dzielimy na:

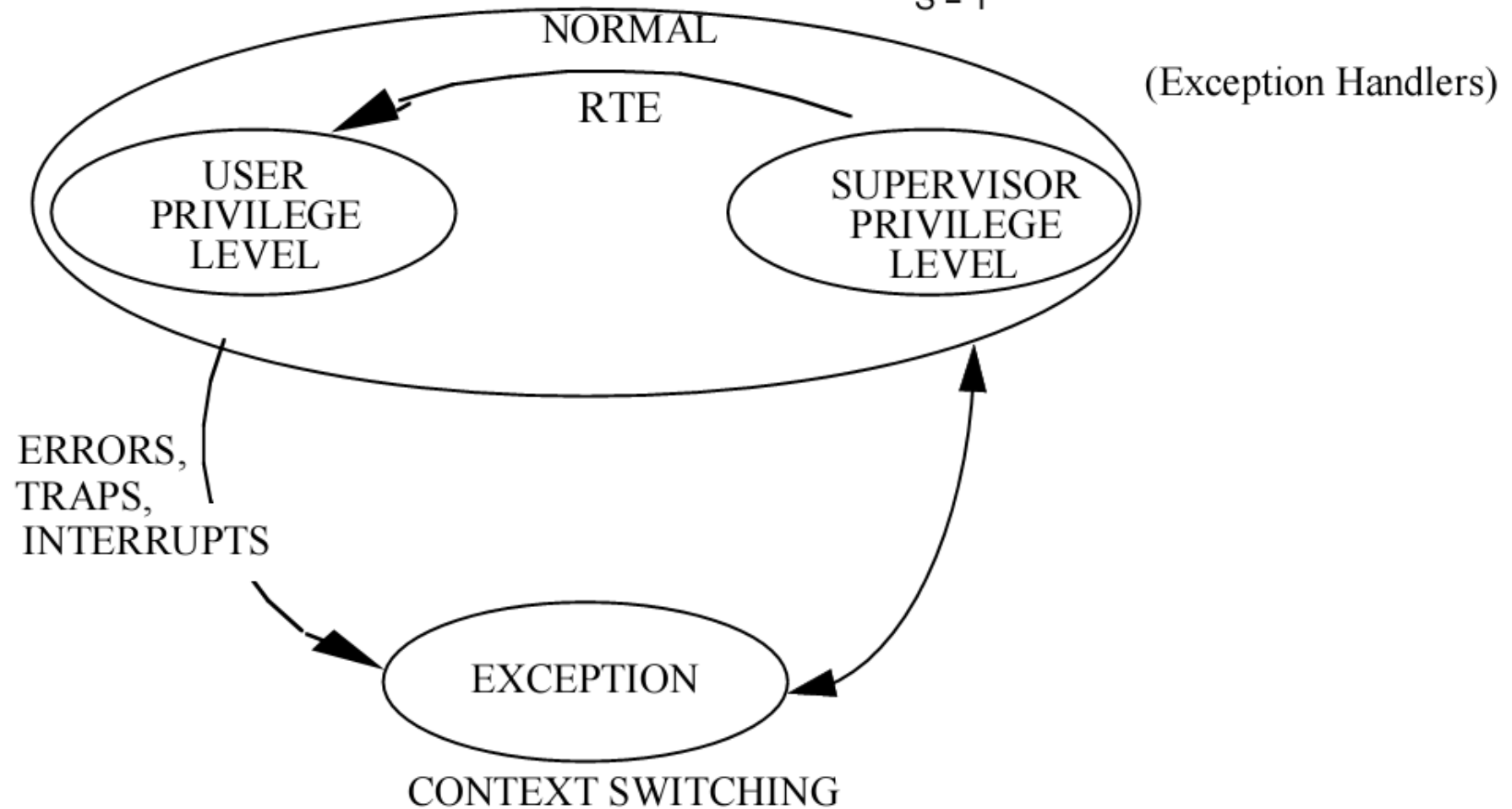
- niepowodzenia (ang. fault)
- błędy nienaprawialne (ang. abort)
- pułapki (ang. trap)

Obsługa sytuacji wyjątkowych

APPLICATIONS

S = 1

OPERATING SYSTEM



Obsługa sytuacji wyjątkowych

Wykonanie niedozwolonej operacji przez procesor w danym stanie uprzywilejowania powoduje wygenerowanie wyjątku.

Obsługa wyjątku obejmuje wszystkie operacje od momentu wykrycia błędu do pobrania pierwszej instrukcji obsługującej sytuację wyjątkową.

1. a) Wykonanie kopii SR,
b) Przejście do trybu superużytkownika ($S \leq 1$; $T \leq 0$; $M \leq 0$),
c) Ustawienie maski IRQ na poziomie zgłaszanego przerwania.
2. Określenie wektora obsługiwanego wyjątku (przerwania).
3. Odłożenie kopii obecnego kontekstu na wierzchołku stosu superużytkownika (ramka zdarzenia wyjątkowego).
4. Obliczenie adresu pierwszej instrukcji procedury obsługującej dany wyjątek (przerwanie).

$$\text{VBR}(31..20) + 4 \times \text{numer_wektora}$$

Detekcja przerwania zostaje wstrzymana na jedną instrukcję przed wykonaniem procedury obsługującej dany wyjątek.

Daje to możliwość wyłączenia przerwania i poprawnego obsłużenia danego wyjątku, np. wykonanie instrukcji **STLDSR**.

STRLDSR

Store/Load Status Register

STRLDSR

First appeared in ISA_C

Operation: If Supervisor State
Then $SP - 4 \rightarrow SP$; zero-filled SR \rightarrow (SP); immediate data \rightarrow SR
Else TRAP

Assembler Syntax: STRLDSR #<data> ← 16 bit

Attributes: Size = word

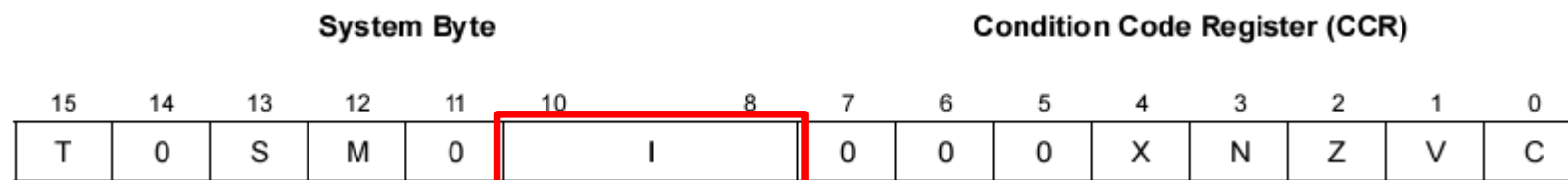
Description: Pushes the contents of the Status Register onto the stack and then reloads the Status Register with the immediate data value. This instruction is intended for use as the first instruction of an interrupt service routine shared across multiple interrupt request levels. It allows the level of the just-taken interrupt request to be stored in memory (using the SR[IML] field), and then masks interrupts by loading the SR[IML] field with 0x7 (if desired). If execution is attempted with bit 13 of the immediate data cleared (attempting to place the processor in user mode), a privilege violation exception is generated. The opcode for STRLDSR is 0x40E7 46FC.

Condition

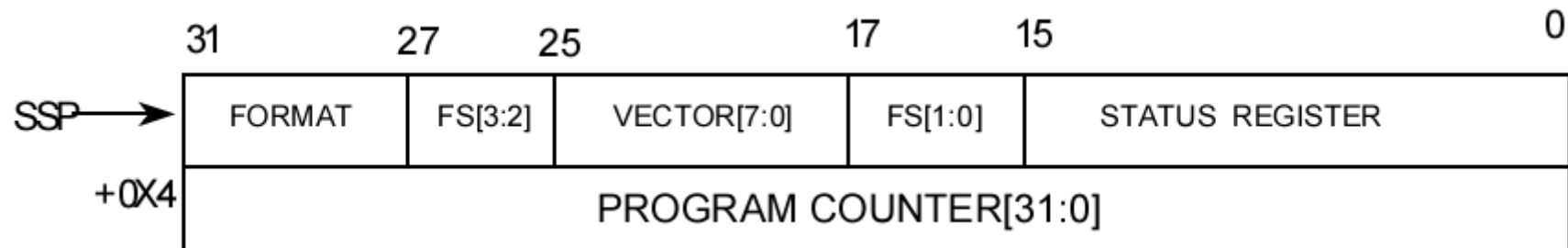
Codes:

X	N	Z	V	C
*	*	*	*	*

- X Set to the value of bit 4 of the immediate operand
- N Set to the value of bit 3 of the immediate operand
- Z Set to the value of bit 2 of the immediate operand
- V Set to the value of bit 1 of the immediate operand
- C Set to the value of bit 0 of the immediate operand



Ramka stosu sytuacji wyjątkowych



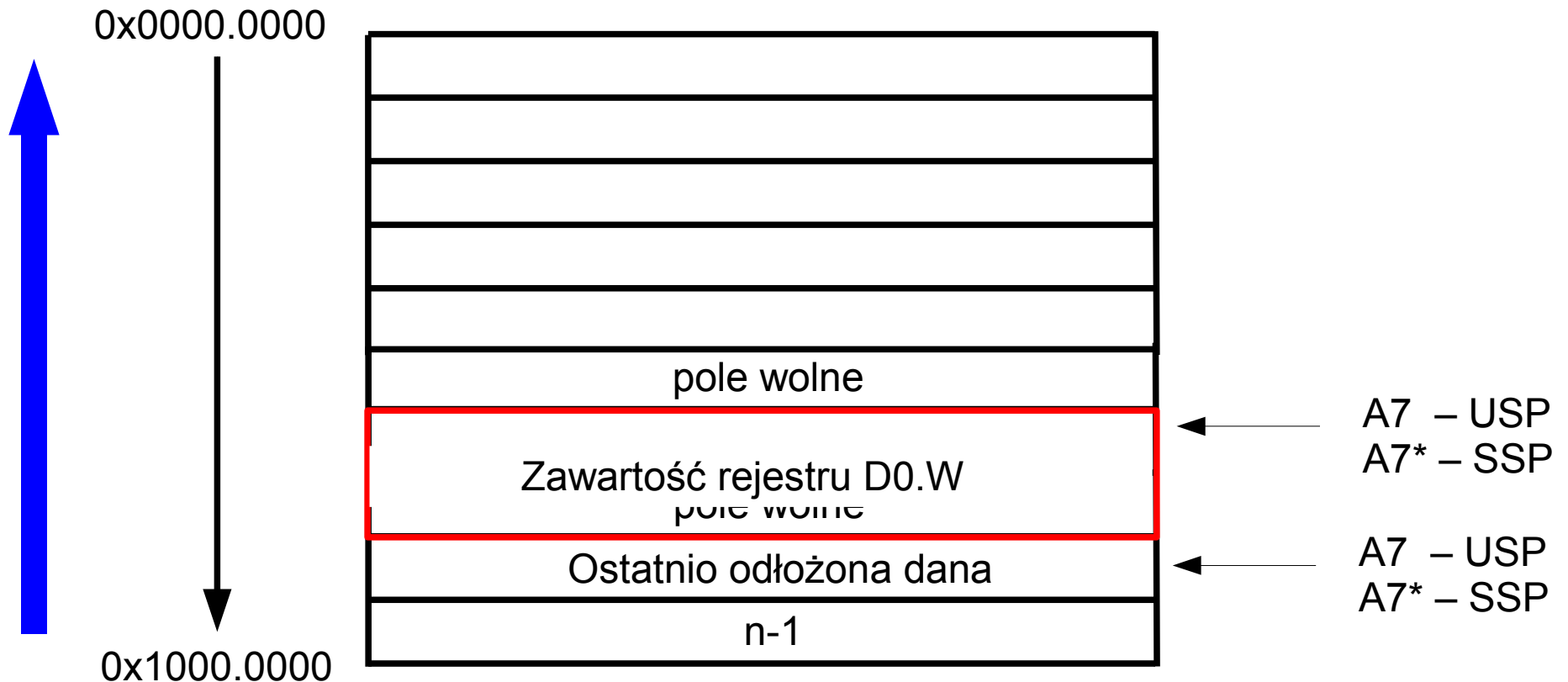
Original SSP @ Time of Exception, Bits 1:0	SSP @ 1st Instruction of Handler	Format Field
00	Original SSP - 8	4
01	Original SSP - 9	5
10	Original SSP - 10	6
11	Original SSP - 11	7



Wskaźnik stosu jest automatycznie wyrównywany do granicy 4 B.

FS[3:0]	Definition
00xx	Reserved
0100	Error on instruction fetch
0101	Reserved
011x	Reserved
1000	Error on operand write
1001	Attempted write to write-protected space
101x	Reserved
1100	Error on operand read
1101	Reserved
111x	Reserved

Stos



MOVE.W %D0, -(%A7) odłożenie zawartości D0.W na stos

MOVEA.L (%A7)+, %A1 zdjęcie danej ze stosu => %A1.L

Aby zapewnić efektywną pracę procesora rejestr A7 powinien wskazywać adres będący wielokrotnością liczby 4.

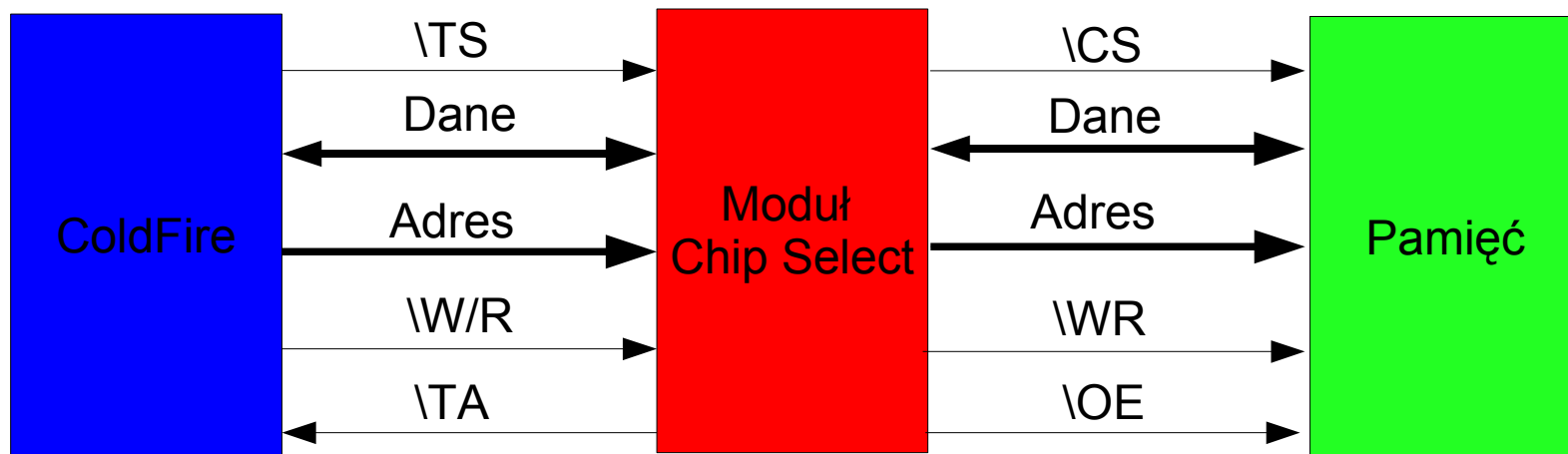
Tablica wektorów wyjątków i przerwań

Vector Number(S)	Vector Offset (Hex)	Stacked Program Counter	Assignment
0	0x000	—	Initial stack pointer
1	0x004	—	Initial program counter
2	0x008	Fault	Access error
3	0x00C	Fault	Address error
4	0x010	Fault	Illegal instruction
5	0x014	Fault	Divide by zero
6–7	0x018–0x01C	—	Reserved
8	0x020	Fault	Privilege violation
9	0x024	Next	Trace
10	0x028	Fault	Unimplemented line-a opcode
11	0x02C	Fault	Unimplemented line-f opcode
12	0x030	Next	Debug interrupt
13	0x034	—	Reserved
14	0x038	Fault	Format error
15–23	0x03C–0x05C	—	Reserved
24	0x060	Next	Spurious interrupt
25–31	0x064–0x07C	—	Reserved
32–47	0x080–0x0BC	Next	Trap # 0-15 instructions
48–63	0x0C0–0x0FC	—	Reserved
64–255	0x100–0x3FC	Next	User-defined interrupts

Wyjątek dostępu do pamięci (1)

- ★ Adres nie istniejącego urządzenia lub pamięci (adres nie jest przypisany do żadnego rejestru konfiguracyjnego CSCR0-CSCR6)
- ★ Próba zapisu do pamięci zabezpieczonej przed zapisem lub tylko do odczytu
- ★ Próba dostępu do pamięci superużytkownika z poziomu zwykłego użytkownika

generują wyjątek **Access Error**



Wyjątek dostępu do pamięci (2)

Odczyt spod chronionego obszaru (SV)

MOVE.W %D0, **-(%A1)**

MOVE.W %D0, **(%A7)+**



Adres obliczony przed
zgłoszeniem wyjątku

Problem podczas wykonywania instrukcji zapisu:

1. MOVE.B #0x12, %D0

2. NOP

wyjątek zgłoszony podczas
wykonywania instrukcji NOP

Wyjątek błędnego adresu

Próba odczytu spod nieparzystego adresu generuje wyjątek
Address Error (CPU 68k)

MOVE.L %D3, **0x3001**

MOVE.W %D0, **-(%A1)**

A1 = 0x800005

MOVE.W %D0, **(%A7)+**

A7 = 0x800005

MOVE.B -8(**%D2**, %D1*2), %D0

indeks w D2 jest 16 bitowy



Scale Factor różny od 1, 2, 4

Wykonanie błędnej instrukcji

Próba wykonania nieznanej instrukcji powoduje wygenerowanie wyjątku **Illegal Instruction**

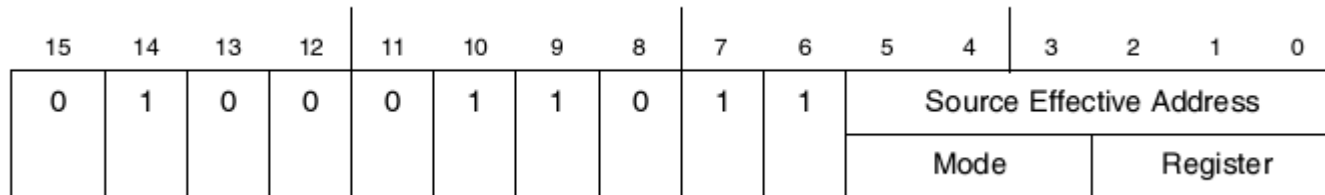
(wyjątek nie dotyczy instrukcji z polem OPCODE 0xAxx lub 0xFxx)



OPeration Code

MOVE.W <ea>y, SR

Instruction Format:



kod rozkazu

tryb adresowanie

operand

Wykonanie instrukcji 0x0000 lub 0x4AFC generuje wyjątek. Instrukcje można wykorzystać jako programowe pułapki.

Wykonanie błędnej instrukcji typu a lub f

Próba wykonania nieznannej instrukcji wykorzystującej jednostkę **eMAC** (Enhanced Multiply and Accumulate) lub **FPU** (Floating Point Unit) powoduje wygenerowanie wyjątku **Unimplemented line-a or line-f opcode**

$ACC + (Ry * Rx)\{\ll | \gg\} SF \rightarrow ACC$

: MAC.sz Ry. {U,L}, Rx. {U,L} SF

Instruction Format:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
1	0	1	0	Register, Rx				0	0	Rx	0	0	Register, Ry			
—	—	—	—	sz	Scale Factor		0	U/Lx	U/Ly	—	—	—	—	—	—	

opcode

Source + FPx → FPx

FADD.fmt <ea>y,FPx
 FADD.D FPy,FPx
 FrADD.fmt <ea>y,FPx
 FrADD.D FPy,FPx

Instruction Format:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	0	0	1	0	0	0	Source Effective Address					
				Mode				Register							
0	R/M	0	Source Specifier				Destination Register, FPx		Opmode						

Dzielenie przez zero

Destination/Source → Destination

DIVS.W <ea>y,Dx 32-bit Dx/16-bit <ea>y Æ (16r:16q) in Dx

DIVS.L <ea>y,Dx 32-bit Dx/32-bit <ea>y Æ 32q in Dx

where q indicates the quotient, and r indicates the remainder

An attempt to divide by zero results in a divide-by-zero exception and no registers are affected. The resulting exception stack frame points to the offending divide opcode. If overflow is detected, the destination register is unaffected. An overflow occurs if the quotient is larger than a 16-bit (.W) or 32-bit (.L) signed integer.

Dzielenie liczby ze znakiem podanej jako operand
przeznaczenia przez liczbę podaną jako operand źródłowy.
Wynik dzielenia umieszczony jest pod adresem
wskazywanym przez operand przeznaczenia.

Jeżeli operand źródłowy jest równy 0 generowany jest
wyjątek **Division by Zero**.

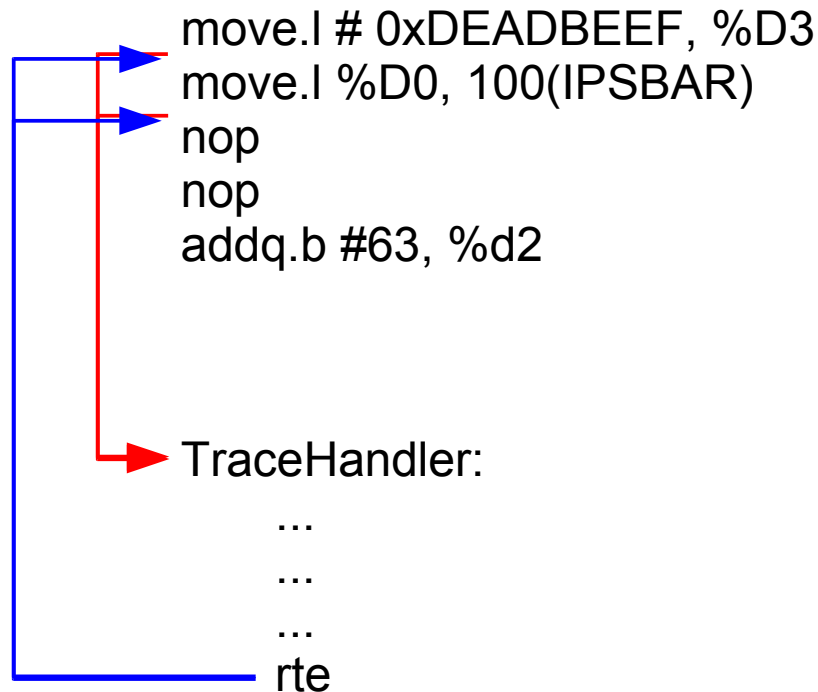
Naruszenie uprawnień superużytkownika

Próba wykonania instrukcji wymagających uprawnień superużytkownika przez użytkownika generuje wyjątek **Privilege Violation**

OPCODE	OPERAND SIZE	ADDRESSING MODE
*HALT	UNSIZED	CONFIGURABLE
MOVE SR	W	Dn
MOVEC	L	Rn,Rc
RTE	UNSIZED	-
STOP	UNSIZED	-
WDEBUG	L	<EA>
*PULSE	UNSIZED	-

Praca krokowa

Jeżeli flaga T w rejestrze SR jest ustawiona generowany jest wyjątek **Trace Exception** (przed rozpoczęciem procedury obsługującej wyjątek, procesor kończy wykonanie bieżącej instrukcji).



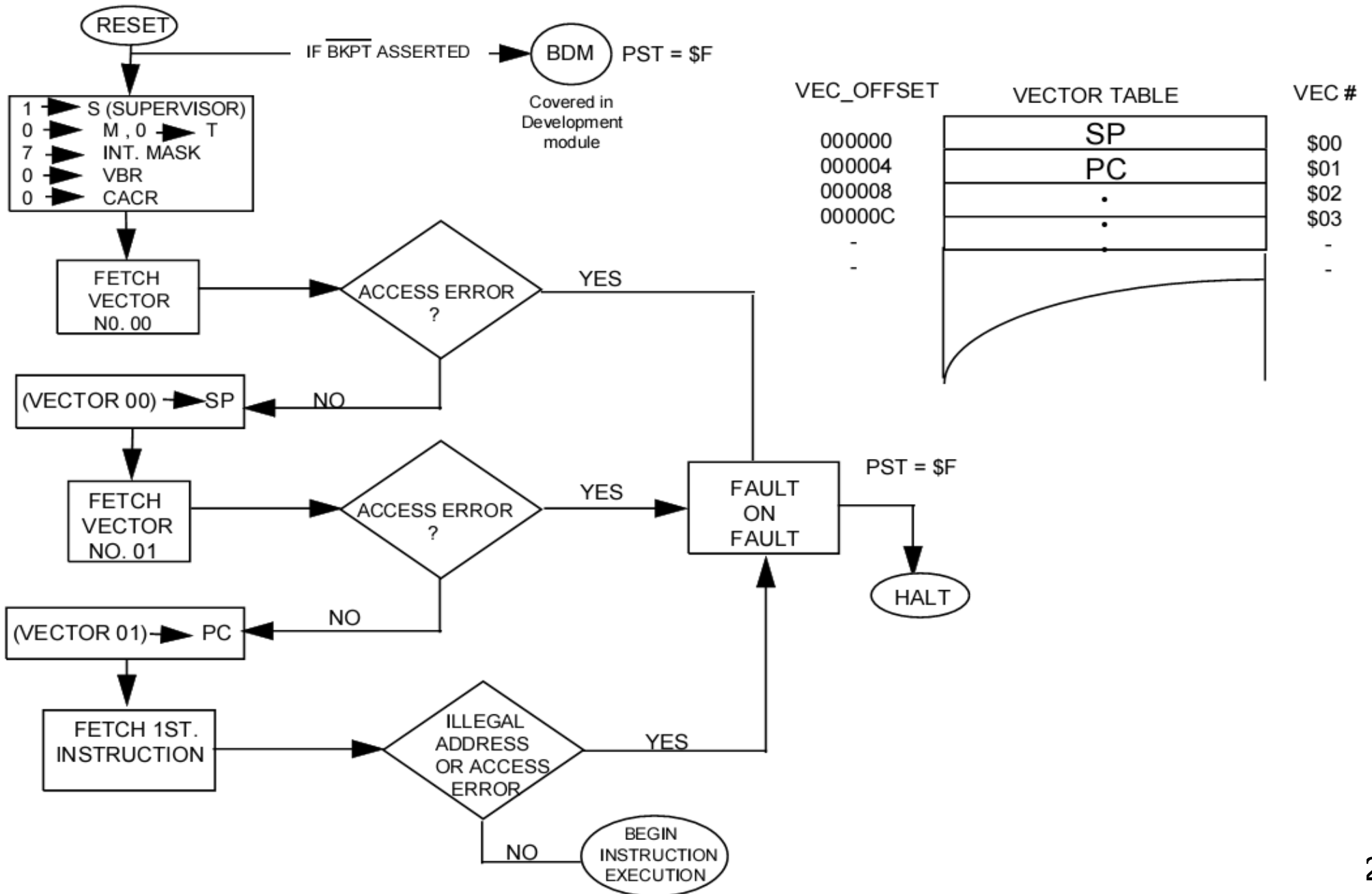
Obsługa nieznanego przerwania

Jeżeli praca procesora zostanie przerwana (w wyniku zgłoszenia przerwania przez INTC0 lub INTC1), jednak sterownik przerwania nie jest w stanie określić źródła przerwania o odpowiednio wysokiej masce następuje wygenerowanie wyjątku **Spurious Interrupt**.

Podwójny wyjątek

Jeżeli podczas obsługi wyjątku zostanie wygenerowany kolejny wyjątek (**fault-on-fault**) procesor natychmiast przerywa pracę. Procesor pozostaje w stanie bezczynności, do momentu wygenerowania sygnału reset.

Obsługa wyjątku reset



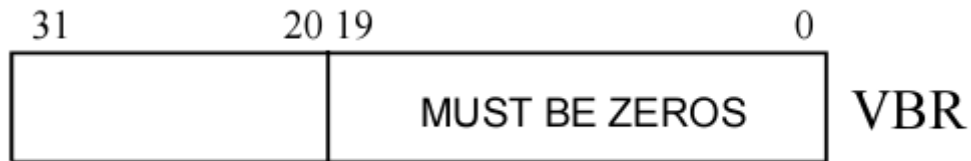
Zawartość rejestru DO po resecie

Bits	Name	Description
31–24	PF	Processor family. This field is fixed to a hex value of 0xCF indicating a ColdFire core is present.
23–20	VER	ColdFire core version number. This field is fixed to a hex value of 0x2 indicating a Version 2 ColdFire core.
19–16	REV	Processor revision number.
15	MAC	MAC execute engine status. Indicates if optional MAC unit is present. 0 MAC execute engine not present in core. (This is the value used for MCF5282.) 1 MAC execute engine is present in core.
14	DIV	Divide execute engine status. Indicates if optional hardware divide unit is present. 0 Divide execute engine not present in core. 1 Divide execute engine is present in core. (This is the value used for MCF5282.)
13	EMAC	EMAC execute engine status. Indicates if optional enhanced MAC unit is present. 0 EMAC execute engine not present in core. 1 EMAC execute engine is present in core. (This is the value used for MCF5282.)
12	FPU	FPU execute engine status. Indicates if optional FPU unit is present. 0 FPU execute engine not present in core. (This is the value used for MCF5282) 1 FPU execute engine is present in core.
11	MMU	Virtual memory management unit status. Indicates if optional MMU unit is present. 0 MMU execute engine not present in core. (This is the value used for MCF5282) 1 MMU execute engine is present in core.
10–8	—	Reserved.
7–4	ISA	Instruction set architecture (ISA) revision number. 0000 ISA_A 0001 ISA_B 0010 ISA_C 1000 ISA_A+ (ISA_A with the addition of the BYTEREV, BITREV, FF1, and STLDSR instructions. This is the value used for MCF5282.) 0x3-0xF Reserved.
3–0	DEBUG	Debug module revision number. 0000 DEBUG_A (This is the value used for MCF5282) 0001 DEBUG_B 0010 DEBUG_C 0011 DEBUG_D 0100 DEBUG_E 0x5-0xF Reserved.

Zawartość rejestru D1 po resecie

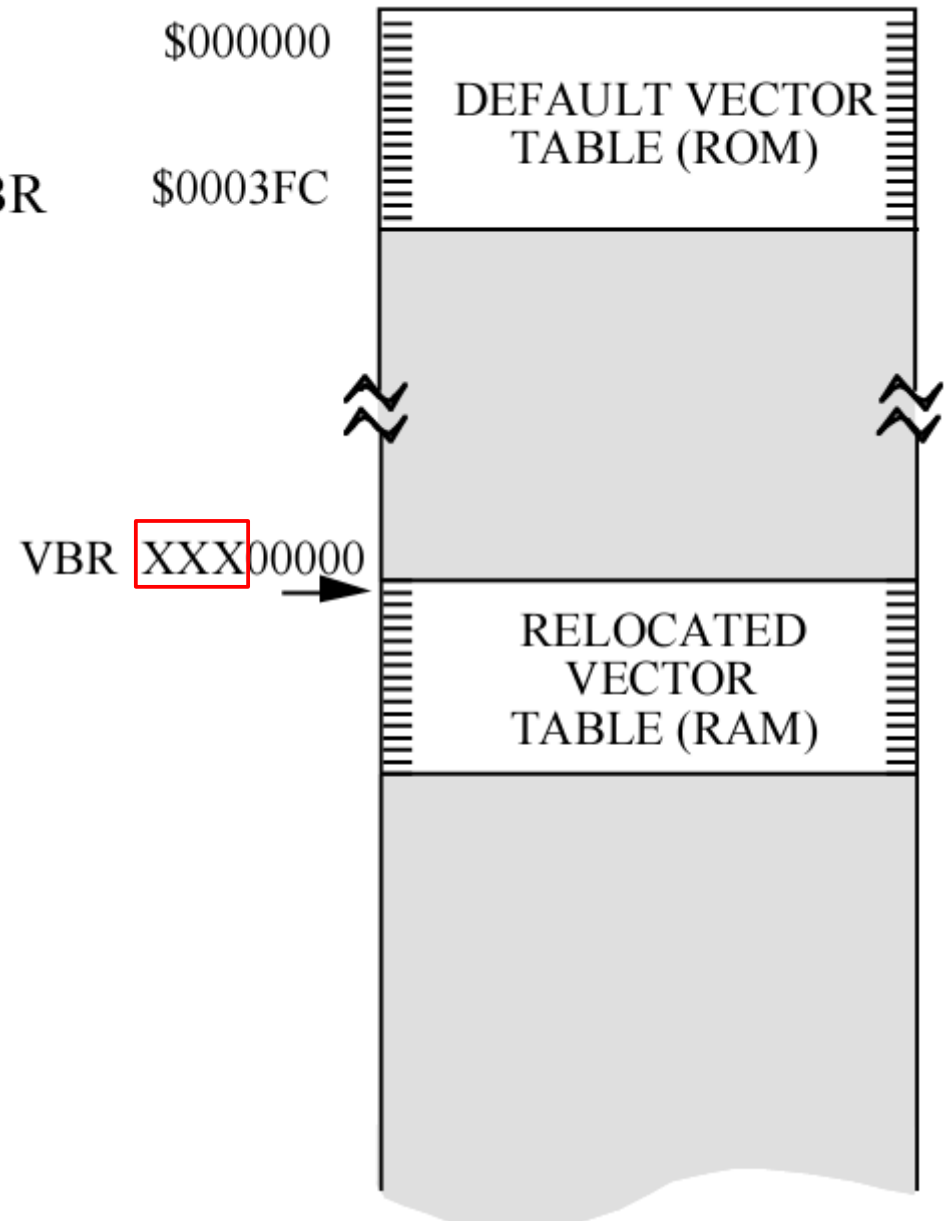
Bits	Name	Description
31–30	CL	Cache line size. This field is fixed to a hex value of 0x0 indicating a 16-byte cache line size.
29–28	ICA	Instruction cache associativity. 00 Four-way. 01 Direct mapped. (This is the value used for MCF5282)
27–24	ICSIZ	Instruction cache size.
23–20	RAM0SIZ	RAM bank 0 size. The first RAM bank can be used for either SRAM or Flash. The first encodings shown are used to indicate the size of a RAM bank, and the second set of encodings indicate the size for a Flash bank. On the MCF5282, RAM0 is associated with the on-chip Flash, so these bits use the Flash encodings.
19–16	ROM0SIZ	ROM bank 0 size.
15–14	BUSW	Encoded bus data width. 00 32-bit data bus (only configuration currently in use).
13–12	DCA	Data cache associativity. 00 Four-way. 01 Direct mapped. (This is the value used for MCF5282)
11–8	DCSIZ	Data cache size.
7–4	RAM1SIZ	RAM bank 1size.
3–0	ROM1SIZ	ROM bank 1size.

Tablica adresów wyjątków oraz przerwań

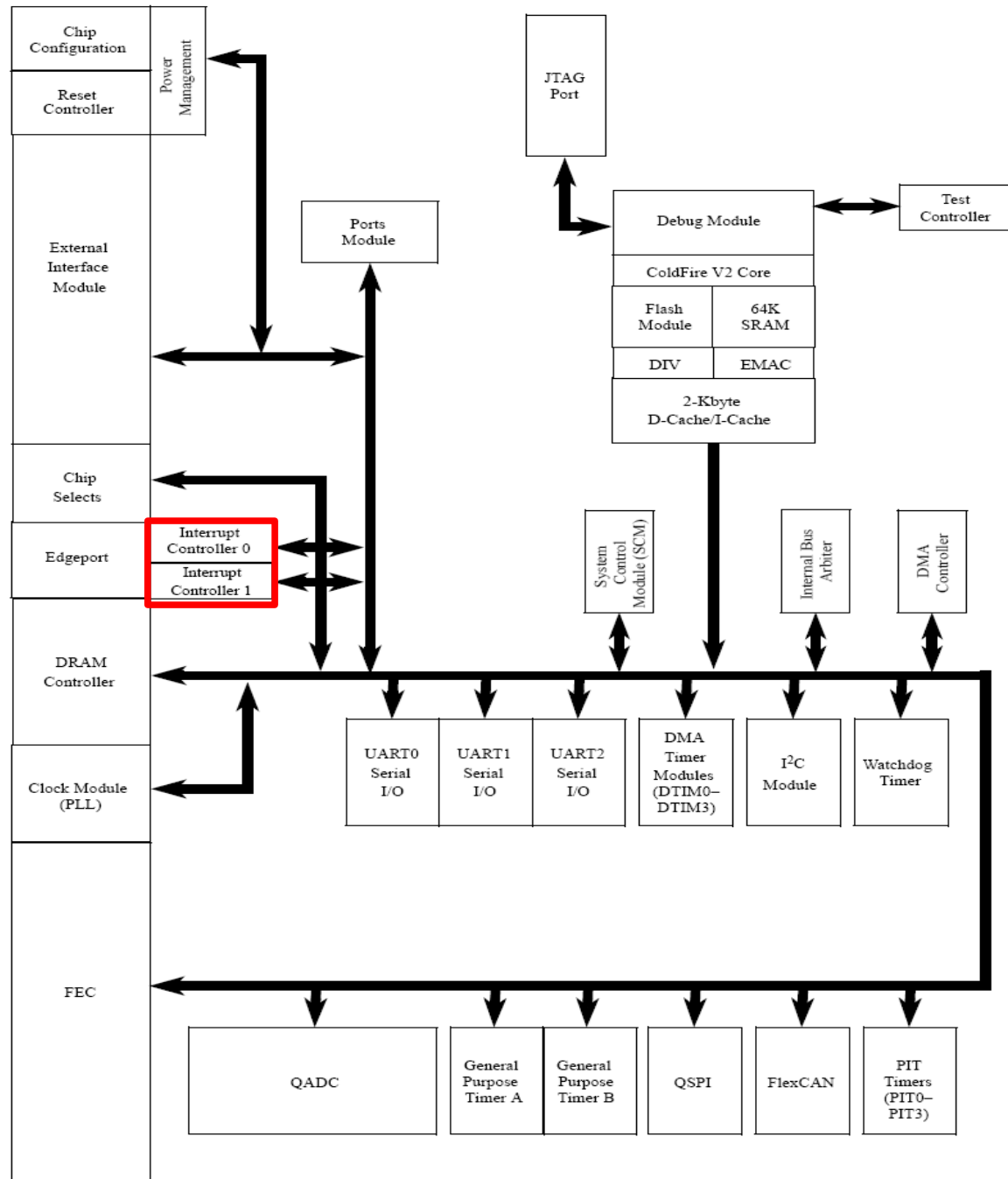


Tablica musi zostać umieszczona na 1 MB granicy.

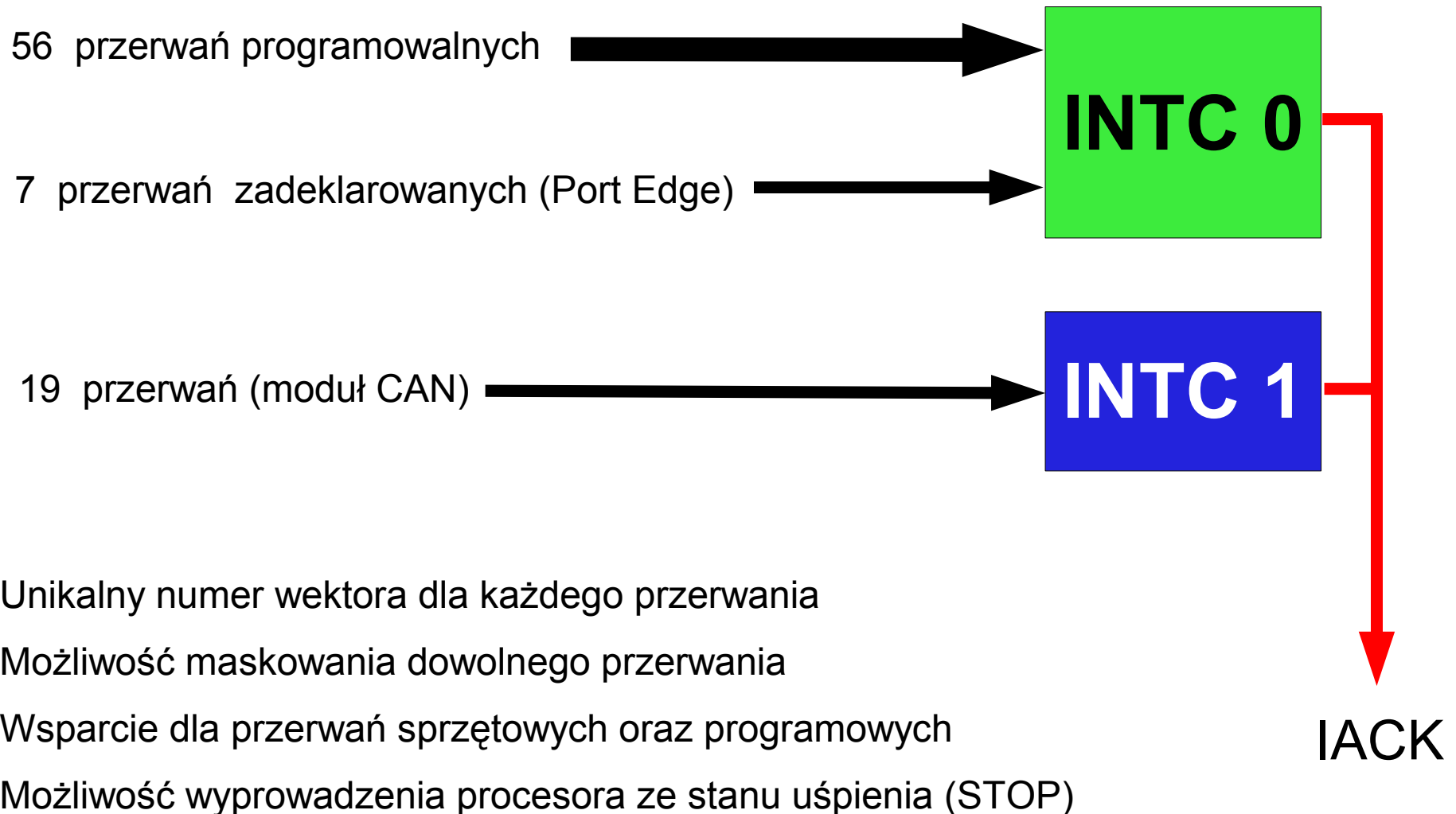
VBR (reset) = 0x0000.0000



MCF5282 – sterownik przerwań



Sterownik przerw (1)



Sterownik przerw (2)

Sterownik przerw rozpoczyna cykl obsługi wyjątku jeżeli zgłoszone przerwanie ma poziom wyższy od poziomu ustawionego w rejestrze SR. Powyższa reguła nie dotyczy przerw o poziomie równym 7.

Przerwania o poziomie równym 7 obsługiwane są zawsze – przerwania niemaskowalne.

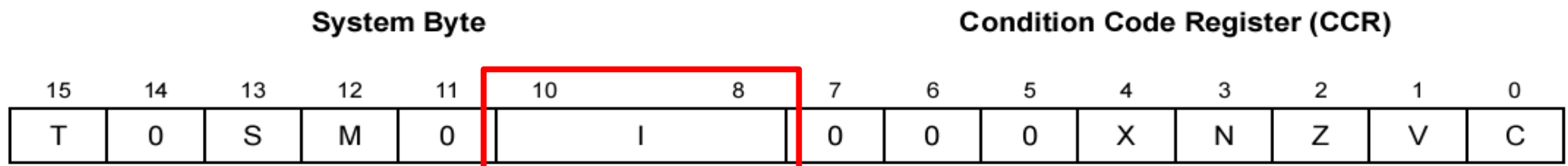
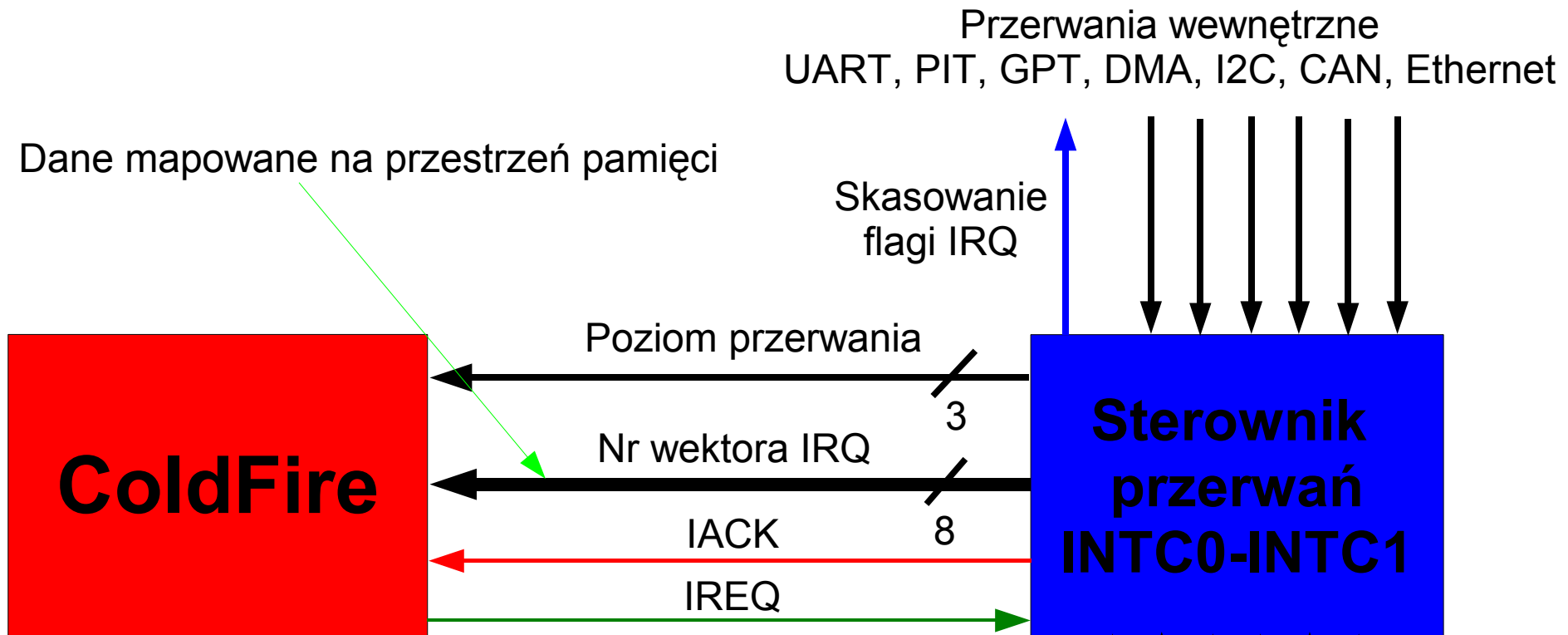


Figure 2-6. Status Register

I > poziom IRQ
IRQ=7 obsługiwane zawsze

Współpraca sterownika przerw z procesorem



Zadania sterownika przerw:

1. Rozpoznanie przerwania
2. Wyznaczenie przerwania o najwyższej masce i priorytecie
3. Obliczenie numeru wektora przerwania

Przerwania zewnętrzne
EPF1-EPF7
poziom/zbocze

Obsługa przerwania

1. Przejście do trybu superużytkownika ($S=1$, $T=0$)
2. Odczytanie wektora aktywnego przerwania
3. Odłożenie na stos kopii rejestru SR, numeru przerwania oraz adresu powrotu
4. Obliczenie przesunięcia w tablicy przerwań oraz odczytanie adresu procedury obsługującej przerwanie
5. Obniżenie maski przerwań w rejestrze SR do poziomu obsługiwanego przerwania
6. Skok do procedury obsługującej przerwanie

Sterownik przerwań

Rejestr stanu SR

15	14	13	12	11	10	8	7	6	5	4	3	2	1	0
T	0	S	M	0	I	0	0	0	0	X	N	Z	V	C

maska przerwań

Priorytety przerwań

ICR[2:0]	Priority	Interrupt Sources
111	7 (Highest)	8-63
110	6	8-63
101	5	8-63
100	4	8-63
—	Fixed Midpoint Priority	1-7
011	3	8-63
010	2	8-63
001	1	8-63
000	0 (Lowest)	8-63

Sterownik przerwań

	Module Offset	Bits[31:24]	Bits[23:16]	Bits[15:8]	Bits[7:0]
R	0x00	Interrupt Pending Register High (IPRH), [63:32]			
	0x04	Interrupt Pending Register Low (IPRL), [31:1]			
	0x08	Interrupt Mask Register High (IMRH), [63:32]			
R/W	0x0c	Interrupt Mask Register Low (IMRL), [31:0]			
	0x10	Interrupt Force Register High (INTFRCH), [63:32]			
R/W	0x14	Interrupt Force Register Low (INTFRCL), [31:1]			
	0x	Interrupt Controller Number		Base Address	
	0x1c	INTC0		IPSBAR + 0xC00	
	0x	INTC1		IPSBAR + 0xD00	
		Global IACK Registers Space ¹		IPSBAR + 0xF00	
	0x80-0x8c	Reserved			
	0xE0	SWIACK	Reserved		
	0xE4	L1IACK	Reserved		
	0xE8	L2IACK	Reserved		
	0xEC	L3IACK	Reserved		
	0xF0	L4IACK	Reserved		
	0xF4	L5IACK	Reserved		
	0xF8	L6IACK	Reserved		
	0xFC	L7IACK	Reserved		

Stan przerwań (1)

Stan przerwań na poszczególnych poziomach **IRLR**

	7	2	1	0
Field	IRQ[7:1]			—
Reset	0000_0000			
R/W	R			
Address	IPSBAR + 0xC18, 0xD18			

Figure 10-7. Interrupt RequestLevel Register (IRLR_n)

Table 10-10. IRQ_n Field Descriptions

Bits	Name	Description
7–1	IRQ	Interrupt requests. Represents the prioritized active interrupts for each level. 0 There are no active interrupts at this level 1 There is an active interrupt at this level
0	—	Reserved

Stan przerwań (2)

Poziom oraz priorytet obsługiwanego

	7	6	4	3	0
Field	—	LEVEL		PRI	
Reset	0000_0000				
R/W	R				
Address	IPSBAR + 0xC19, 0xD19				

Figure 10-8. IACK Level and Priority Register (IACKLPR_n)

Table 10-11. IACKLPR_n Field Descriptions

Bits	Name	Description
7	—	Reserved
6–4	LEVEL	Interrupt level. Represents the interrupt level currently being acknowledged.
3–0	PRI	Interrupt Priority. Represents the priority within the interrupt level of the interrupt currently being acknowledged. 0 Priority 0 1 Priority 1 2 Priority 2 3 Priority 3 4 Priority 4 5 Priority 5 6 Priority 6 7 Priority 7 8 Mid-Point Priority associated with the fixed level interrupts only

Rejstry konfiguracyjne (1)

Rejstry konfiguracyjne ICR0.01 - ICR1.63

	7	6	5	3	2	0
Field	—		IL		IP	
Reset	0000_0000					
R/W	R/W (Read only for ICR _n 1-ICR _n 7)					
Address	See Table 10-2 and Table 10-3 for register offsets					

Figure 10-9. Interrupt Control Register (ICR_nx)

Table 10-12. ICR_nx Field Descriptions

Bits	Name	Description
7–6	—	Reserved, should be cleared.
5–3	IL	Interrupt level. Indicates the interrupt level assigned to each interrupt input.
2–0	IP	Interrupt priority. Indicates the interrupt priority for internal modules within the interrupt-level assignment. 000b represents the lowest priority and 111b represents the highest. For the fixed level interrupt sources, the priority is fixed at the midpoint for the level, and the IP field will always read as 000b.

Rejestry diagnostyczne

Module Offset	Bits[31:24]	Bits[23:16]	Bits[15:8]	Bits[7:0]
0xE0	SWIACK		Reserved	
0xE4	L1IACK		Reserved	
0xFC	L7IACK		Reserved	

Table 10-15. SWIACK and L1IACK-L7IACK Field Descriptions

Bits	Name	Description
7-0	VECTOR	Vector number. A read from the SWIACK register returns the vector number associated with the highest level, highest priority unmasked interrupt source. A read from one of the LnACK registers returns the highest priority unmasked interrupt source within the level.

Rejestry SWIACK oraz LnIACK ułatwiają realizację programowych przerwaniań.
Analiza bitu M rejestru statusowego.

Źródła przerw sterownika INTC0 (1)

Source	Module	Flag	Source Description	Flag Clearing Mechanism
1	EPORT	EPF1	Edge port flag 1	Write EPF1 = 1
2		EPF2	Edge port flag 2	Write EPF2 = 1
3		EPF3	Edge port flag 3	Write EPF3 = 1
4		EPF4	Edge port flag 4	Write EPF4 = 1
5		EPF5	Edge port flag 5	Write EPF5 = 1
6		EPF6	Edge port flag 6	Write EPF6 = 1
7		EPF7	Edge port flag 7	Write EPF7 = 1
8	SCM	SWT1	Software watchdog timeout	Cleared when service complete
9	DMA	DONE	DMA Channel 0 transfer complete	Write DONE = 1
10		DONE	DMA Channel 1 transfer complete	Write DONE = 1
11		DONE	DMA Channel 2 transfer complete	Write DONE = 1
12		DONE	DMA Channel 3 transfer complete	Write DONE = 1
13	UART0	Multiple	UART0 interrupt	Cleared when service complete
14	UART1	Multiple	UART1 interrupt	Cleared when service complete
15	UART2	Multiple	UART2 interrupt	Cleared when service complete
17	I ² C	IIF	I ² C interrupt	Write IIF = 0
18	QSPI	Multiple	QSPI interrupt	See QIR description
19	DTIM0	CAP/REF	DTIM0 capture/reference event	Write CAP = 1 or REF = 1
20	DTIM1	CAP/REF	DTIM1 capture/reference event	Write CAP = 1 or REF = 1
21	DTIM2	CAP/REF	DTIM2 capture/reference event	Write CAP = 1 or REF = 1
22	DTIM3	CAP/REF	DTIM3 capture/reference event	Write CAP = 1 or REF = 1
23	FEC	X_INTF	Transmit frame interrupt	Write X_INTF = 1
24		X_INTB	Transmit buffer interrupt	Write X_INTB = 1
34		BABT	Babbling transmit error	Write BABT = 1
35		BABR	Babbling receive error	Write BABR = 1
36	PMM	LVDF	LVD	Write LVDF = 1

Źródła przerwania sterownika INTC0 (2)

Source	Module	Flag	Source Description	Flag Clearing Mechanism
37	QADC	CF1	Queue 1 conversion complete	Write CF1 = 0 after reading CF1 = 1
38		CF2	Queue 2 conversion complete	Write CF2 = 0 after reading CF2 = 1
39		PF1	Queue 1 conversion pause	Write PF1 = 0 after reading PF1 = 1
40		PF2	Queue 2 conversion pause	Write PF2 = 0 after reading PF2 = 1
41	GPTA	TOF	Timer overflow	Write TOF = 1 or access TIMCNTH/L if TFFCA = 1
42		PAIF	Pulse accumulator input	Write PAIF = 1 or access PAC if TFFCA = 1
43		PAOVF	Pulse accumulator overflow	Write PAOVF = 1 or access PAC if TFFCA = 1
44		C0F	Timer channel 0	Write C0F = 1 or access IC/OC if TFFCA = 1
45		C1F	Timer channel 1	Write C1F = 1 or access IC/OC if TFFCA = 1
46		C2F	Timer channel 2	Write C2F = 1 or access IC/OC if TFFCA = 1
47		C3F	Timer channel 3	Write C3F = 1 or access IC/OC if TFFCA = 1
48	GPTB	TOF	Timer overflow	Write TOF = 1 or access TIMCNTH/L if TFFCA = 1
49		PAIF	Pulse accumulator input	Write PAIF = 1 or access PAC if TFFCA = 1
50		PAOVF	Pulse accumulator overflow	Write PAOVF = 1 or access PAC if TFFCA = 1
51		C0F	Timer channel 0	Write C0F = 1 or access IC/OC if TFFCA = 1
52		C1F	Timer channel 1	Write C1F = 1 or access IC/OC if TFFCA = 1
53		C2F	Timer channel 2	Write C2F = 1 or access IC/OC if TFFCA = 1
54		C3F	Timer channel 3	Write C3F = 1 or access IC/OC if TFFCA = 1
55	PIT0	PIF	PIT interrupt flag	Write PIF = 1 or write PMR
56	PIT1	PIF	PIT interrupt flag	Write PIF = 1 or write PMR
57	PIT2	PIF	PIT interrupt flag	Write PIF = 1 or write PMR
58	PIT3	PIF	PIT interrupt flag	Write PIF = 1 or write PMR
59	CFM	CBEIF	SGFM buffer empty	Write CBEIF = 1
60	CFM	CCIF	SGFM command complete	Cleared automatically
61	CFM	PVIF	Protection violation	Cleared automatically
62	CFM	AEIF	Access error	Cleared automatically

Źródła przerw sterownika INTC1

Source	Module	Flag	Source Description	Flag Clearing Mechanism
1-7	Not Used			
8	FLEX CAN	BUF0I	Message buffer 0 interrupt	Write BUF0I = 1 after reading BUF0I = 1
9		BUF1I	Message buffer 1 interrupt	Write BUF1I = 1 after reading BUF1I = 1
10		BUF2I	Message buffer 2 interrupt	Write BUF2I = 1 after reading BUF2I = 1
11		BUF3I	Message buffer 3 interrupt	Write BUF3I = 1 after reading BUF3I = 1
12		BUF4I	Message buffer 4 interrupt	Write BUF4I = 1 after reading BUF4I = 1
13		BUF5I	Message buffer 5 interrupt	Write BUF5I = 1 after reading BUF5I = 1
14		BUF6I	Message buffer 6 interrupt	Write BUF6I = 1 after reading BUF6I = 1
15		BUF7I	Message buffer 7 interrupt	Write BUF7I = 1 after reading BUF7I = 1
16		BUF8I	Message buffer 8 interrupt	Write BUF8I = 1 after reading BUF8I = 1
17		BUF9I	Message buffer 9 interrupt	Write BUF9I = 1 after reading BUF9I = 1
18		BUF10I	Message buffer 10 interrupt	Write BUF10I = 1 after reading BUF10I = 1
19		BUF11I	Message buffer 11 interrupt	Write BUF11I = 1 after reading BUF11I = 1
20		BUF12I	Message buffer 12 interrupt	Write BUF12I = 1 after reading BUF12I = 1
21		BUF13I	Message buffer 13 interrupt	Write BUF13I = 1 after reading BUF13I = 1
22		BUF14I	Message buffer 14 interrupt	Write BUF14I = 1 after reading BUF14I = 1
23		BUF15I	Message buffer 15 interrupt	Write BUF15I = 1 after reading BUF15I = 1
24		ERR_INT	Error interrupt	Read error bits in ESR or write ERR_INT = 0
25		BOFF_INT	Bus-off interrupt	Write BOFF_INT = 0
26		WAKE_INT	Wake-up interrupt	Write WAKE_INT = 0
27-63	Not used			

Procedura obsługi przerwania w asemblerze

1. Konfiguracja modułów
 2. Skopiowanie tablicy wektorów do pamięci RAM
 3. MOVE.L #Adres_tablicy, %A0 | adres tablicy => A0, (na granicy 1 MB)
 4. MOVEC.L %A0,%VBR | A0 => VBR
 5. LEA IRQ_handler, %A1 | ładuj adres handlera do A1
 6. MOVE.L %A1, 4*77(%A0) | prześlij A1 pod adres wskazywany przez A0
| UART0 = (64+13) x 4
 7. Ustaw priorytet oraz poziom przerwania ICR13 = 0x-- 100 111 (level/priority)
 8. Włącz przerwania
- Program główny -----
-

IRQ_handler:

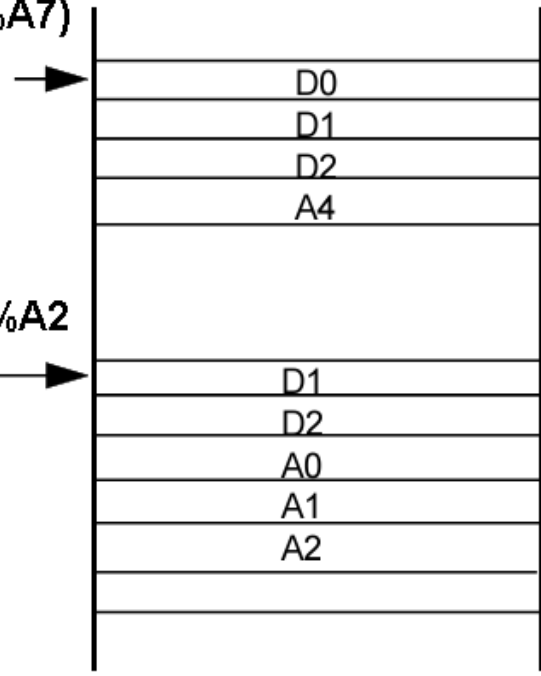
1. STRLDSR #0x2700 | odłóż SR na stos i załaduj nową wart. 0x2700
| IRQ są wyłączone SR[I] = 7
 2. LEA -24(%A7), %A7 | zarezerwuj miejsce na stosie, 2(0/1) execution
MOVEM.L %D0-%D2/%A0-%A2,(%A7) | odłóż rejestry na stos, 1+n(0/n) execution
 3. | Skasuj flagę IRQ (Timer DMA => done = 1)
- Program obsługujący przerwanie -----

5. MOVEM.L (%A7),%D0-%D2/%A0-%A2 | zdejmij rejestry ze stosu, 1+n(n/0) execution
LEA 24(%A7), %A7 | zwolnij miejsce na stosie, 2(1/0) execution
MOVE.W (%A7)+,%D0 | odzyskaj oryginalny SR.W ze stosu
MOVE.W %D0,%SR | D0 => SR
5. RTE | powrót do programu głównego

Instrukcja specjalna MOVEM (1)

INSTRUCTION	SIZE	OPERATION
MOVEM	L	REGISTER LIST \longrightarrow DEST. REGS. SOURCE \longrightarrow

MOVEM.L %D0-%D2/%A4, (%A7)

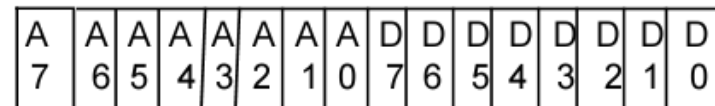


MOVEM.L (%A7), %D1-%D2/%A0-%A2

SP \longrightarrow

Dozwolone tryby adresowania:

- (An)
- (d16, An)



Instrukcja specjalna MOVEM (2)

