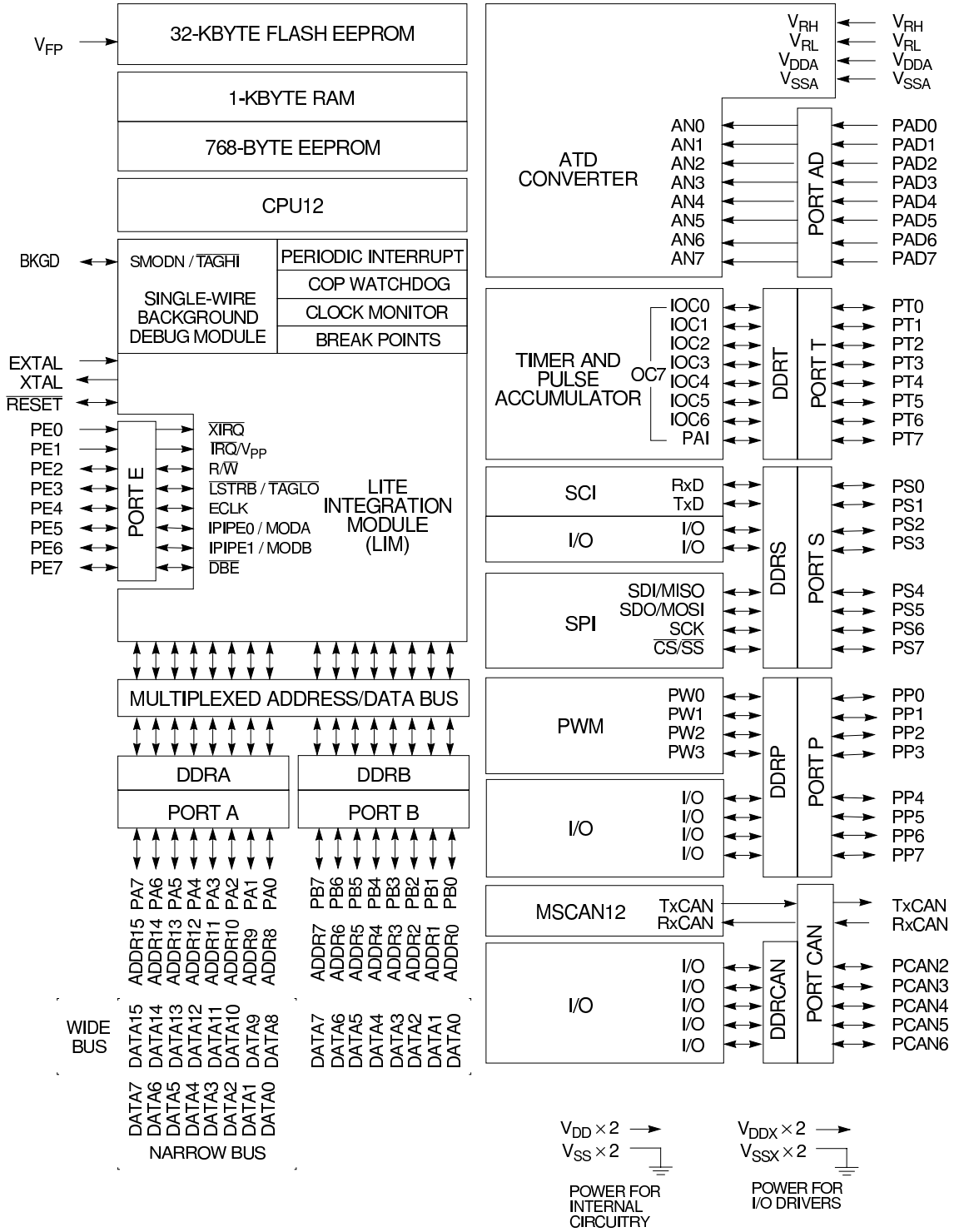
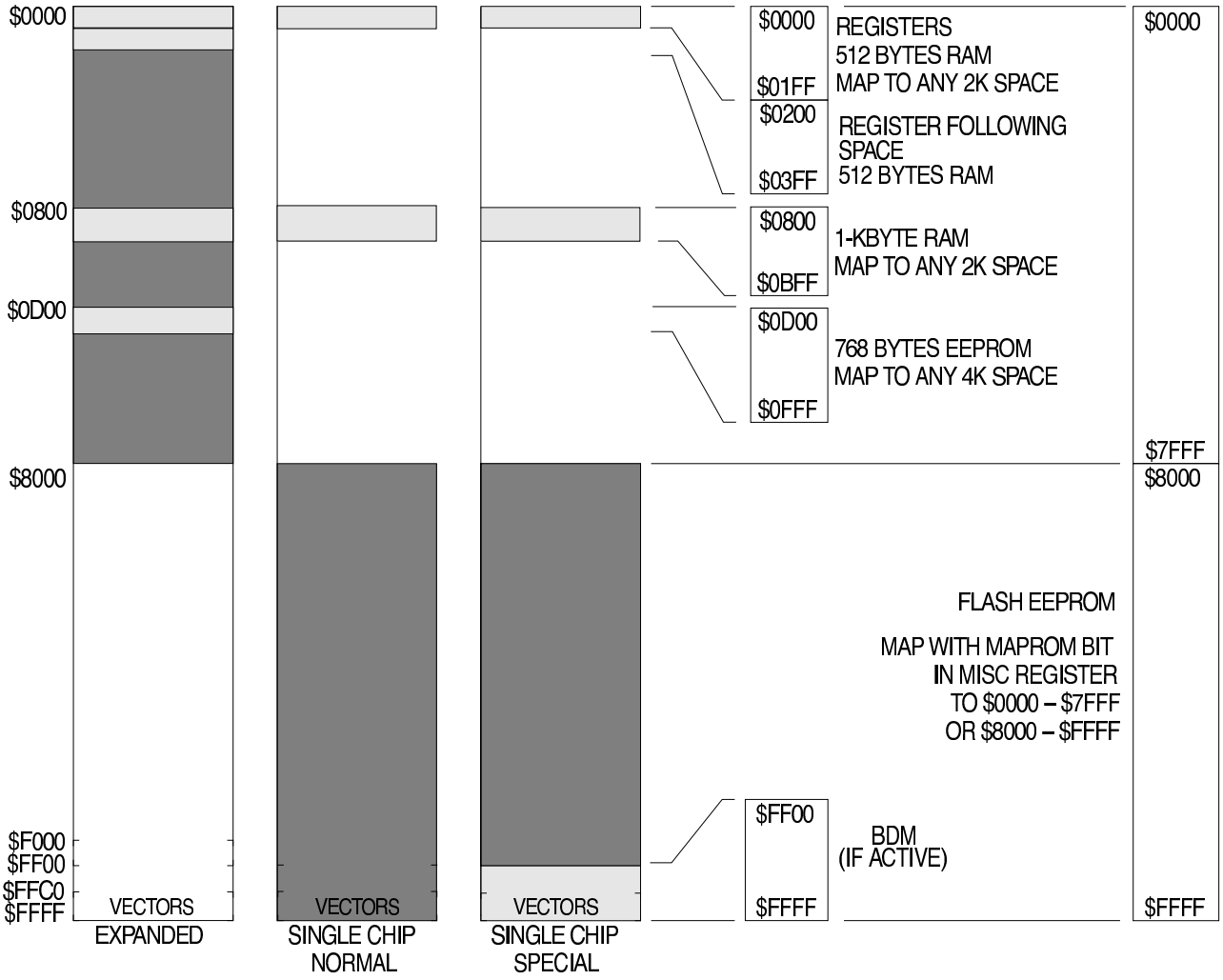
Struktura mikrokontrolera MC68HC12BC32

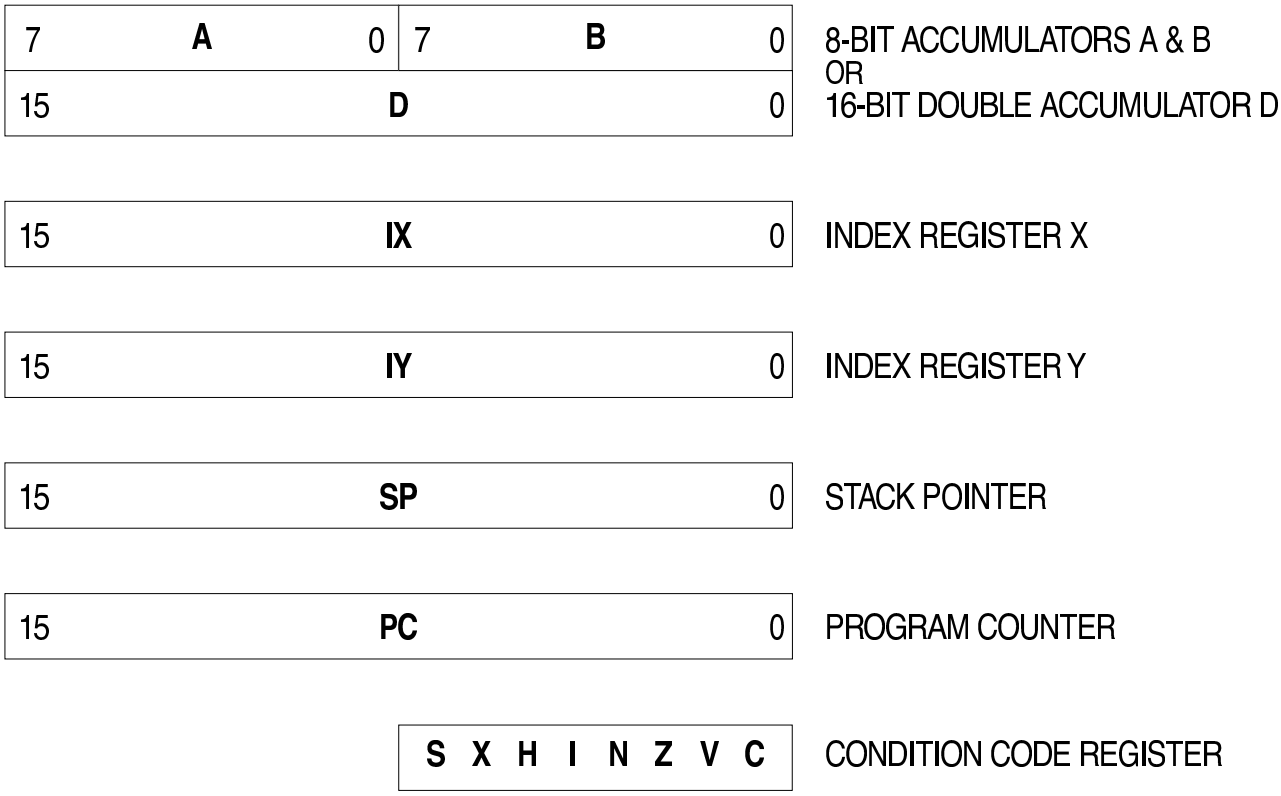


Mapa pamięci i tryby pracy MC68HC12BC32



BKGD	MODB	MODA	Mode	Port A	Port B
0	0	0	Special Single Chip	General Purpose I/O	General Purpose I/O
0	0	1	Special Expanded Narrow	ADDR[15:8]/DATA[7:0]	ADDR[7:0]
0	1	0	Special Peripheral	ADDR/DATA	ADDR/DATA
0	1	1	Special Expanded Wide	ADDR/DATA	ADDR/DATA
1	0	0	Normal Single Chip	General Purpose I/O	General Purpose I/O
1	0	1	Normal Expanded Narrow	ADDR[15:8]/DATA[7:0]	ADDR[7:0]
1	1	0	Reserved (Forced to Peripheral)	—	—
1	1	1	Normal Expanded Wide	ADDR/DATA	ADDR/DATA

Rejestry CPU12



Tryby adresowania CPU12

Addressing Mode	Source Format	Abbreviation	Description
Inherent	**INST** (no externally supplied operands)	INH	Operands (if any) are in CPU registers
Immediate	**INST #***opr8i* or **INST #***opr16i*	IMM	Operand is included in instruction stream 8- or 16-bit size implied by context
Direct	**INST** *opr8a*	DIR	Operand is the lower 8-bits of an address in the range $0000 – $00FF
Extended	**INST** *opr16a*	EXT	Operand is a 16-bit address
Relative	**INST** *rel8* or **INST** *rel16*	REL	An 8-bit or 16-bit relative offset from the current pc is supplied in the instruction
Indexed (5-bit offset)	**INST** *oprx5,xysp*	IDX	5-bit signed constant offset from x, y, sp, or pc
Indexed (auto pre-decrement)	**INST** *oprx3,–xys*	IDX	Auto pre-decrement x, y, or sp by 1 ~ 8
Indexed (auto pre-increment)	**INST** *oprx3,+xys*	IDX	Auto pre-increment x, y, or sp by 1 ~ 8
Indexed (auto post-decrement)	**INST** *oprx3,xys–*	IDX	Auto post-decrement x, y, or sp by 1 ~ 8
Indexed (auto post-increment)	**INST** *oprx3,xys+*	IDX	Auto post-increment x, y, or sp by 1 ~ 8
Indexed (accumulator offset)	**INST** *abd,xysp*	IDX	Indexed with 8-bit (A or B) or 16-bit (D) accumulator offset from x, y, sp, or pc
Indexed (9-bit offset)	**INST** *oprx9,xysp*	IDX1	9-bit signed constant offset from x, y, sp, or pc (lower 8-bits of offset in one extension byte)
Indexed (16-bit offset)	**INST** *oprx16,xysp*	IDX2	16-bit constant offset from x, y, sp, or pc (16-bit offset in two extension bytes)
Indexed-Indirect (16-bit offset)	**INST [***oprx16,xysp***]**	[IDX2]	Pointer to operand is found at... 16-bit constant offset from x, y, sp, or pc (16-bit offset in two extension bytes)
Indexed-Indirect (D accumulator offset)	**INST [D,***xysp***]**	[D,IDX]	Pointer to operand is found at... x, y, sp, or pc plus the value in D

Rozszerzenia trybów indeksowych w CPU12

Postbyte Code (xb)	Source Code Syntax	Comments rr; 00 = X, 01 = Y, 10 = SP, 11 = PC
rr0nnnnn	,r n,r –n,r	**5-bit constant offset** n = –16 to +15 r can specify X, Y, SP, or PC
111rr0zs	n,r –n,r	**Constant offset** (9- or 16-bit signed) z- 0 = 9-bit with sign in LSB of postbyte(s) -256 < n < 255 1 = 16-bit 0 < n < 65,535 if z = s = 1, 16-bit offset indexed-indirect (see below) r can specify X, Y, SP, or PC
111rr011	[n,r]	**16-bit offset indexed-indirect** rr can specify X, Y, SP, or PC 0 < n < 65,535
rr1pnnnn	n,–r n,+r n,r– n,r+	**Auto pre-decrement/increment** or **Auto post-decrement/increment**; p = pre-(0) or post-(1), n = –8 to –1, +1 to +8 r can specify X, Y, or SP (PC not a valid choice) +8 = 0111 … +1 = 0000 -1 = 1111 … -8 = 1000
111rr1aa	A,r B,r D,r	**Accumulator offset** (unsigned 8-bit or 16-bit) aa- 00 = A 01 = B 10 = D (16-bit) 11 = see accumulator D offset indexed-indirect r can specify X, Y, SP, or PC
111rr111	[D,r]	**Accumulator D offset indexed-indirect** r can specify X, Y, SP, or PC

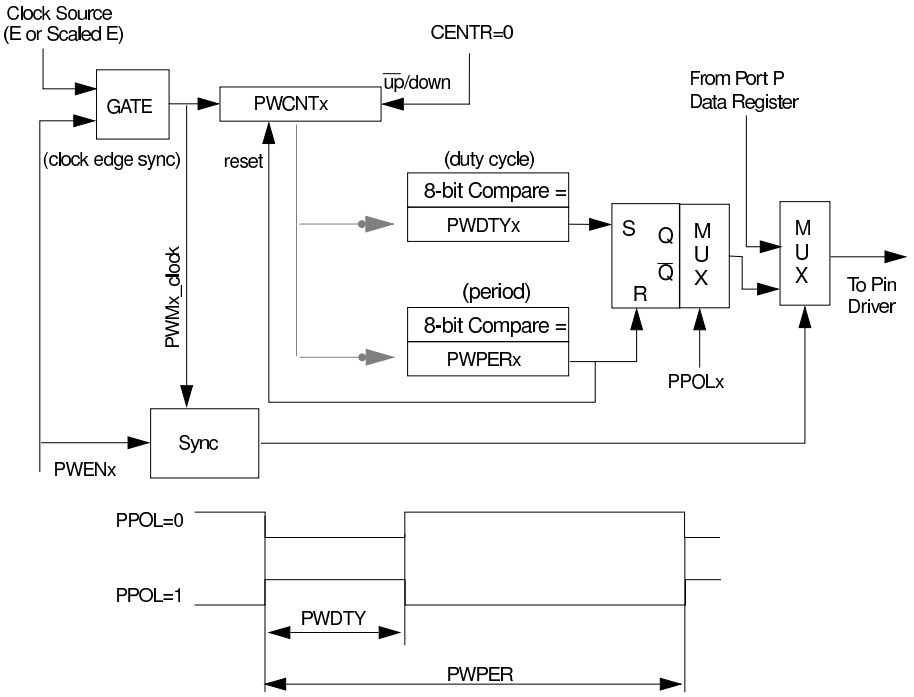
Wektory przerwań 68HC12

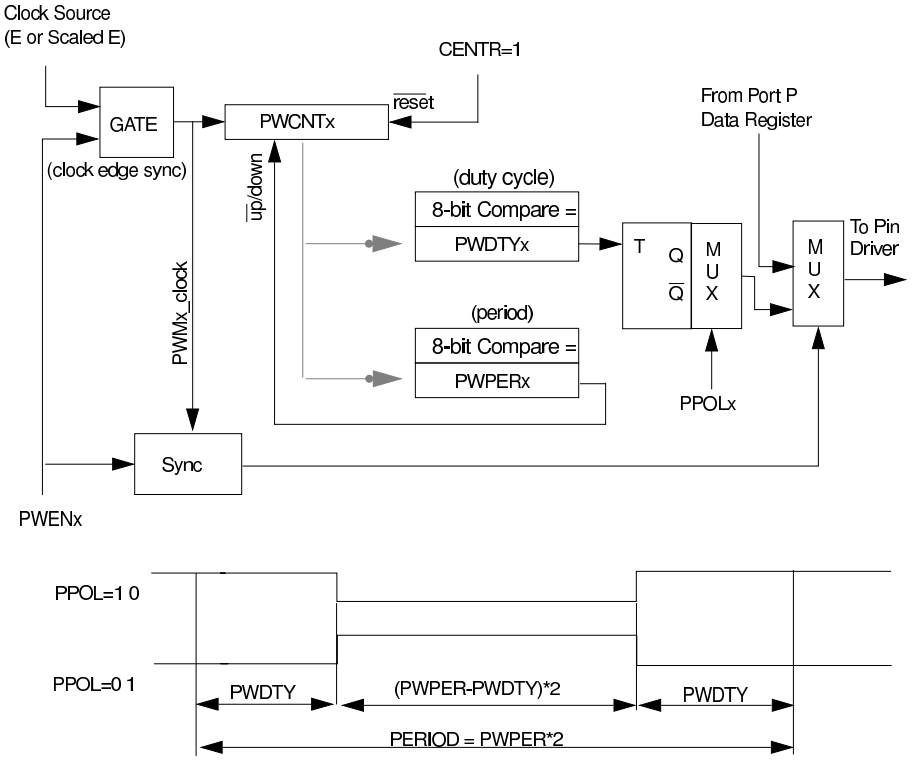
Vector Address	Interrupt Source	CCR Mask	Local Enable Register (Bit)	HPRIO Value to Elevate
$FFFE, $FFFF	Reset	none	none	–
$FFFC, $FFFD	COP Clock Monitor Fail Reset	none	COPCTL (CME, FCME)	–
$FFFA, $FFFB	COP Failure Reset	none	COP rate selected	–
$FFF8, $FFF9	Unimplemented Instruction Trap	none	none	–
$FFF6, $FFF7	SWI	none	none	–
$FFF4, $FFF5	XIRQ	X bit	none	–
$FFF2, $FFF3	IRQ	I bit	INTCR (IRQEN)	$F2
$FFF0, $FFF1	Real Time Interrupt	I bit	RTICTL (RTIE)	$F0
$FFEE, $FFEF	Timer Channel 0	I bit	TMSK1 (C0I)	$EE
$FFEC, $FFED	Timer Channel 1	I bit	TMSK1 (C1I)	$EC
$FFEA, $FFEB	Timer Channel 2	I bit	TMSK1 (C2I)	$EA
$FFE8, $FFE9	Timer Channel 3	I bit	TMSK1 (C3I)	$E8
$FFE6, $FFE7	Timer Channel 4	I bit	TMSK1 (C4I)	$E6
$FFE4, $FFE5	Timer Channel 5	I bit	TMSK1 (C5I)	$E4
$FFE2, $FFE3	Timer Channel 6	I bit	TMSK1 (C6I)	$E2
$FFE0, $FFE1	Timer Channel 7	I bit	TMSK1 (C7I)	$E0
$FFDE, $FFDF	Timer Overflow	I bit	TMSK2 (TOI)	$DE
$FFDC, $FFDD	Pulse Accumulator Overflow	I bit	PACTL (PAOVI)	$DC
$FFDA, $FFDB	Pulse Accumulator Input Edge	I bit	PACTL (PAI)	$DA
$FFD8, $FFD9	SPI Serial Transfer Complete	I bit	SP0CR1 (SPIE)	$D8
$FFD6, $FFD7	SCI 0	I bit	SC0CR2 (TIE, TCIE, RIE, ILIE)	$D6
$FFD4, $FFD5	Reserved	I bit		$D4
$FFD2, $FFD3	ATD	I bit	ATDCTL2 (ASCIE)	$D2
$FFD0, $FFD1	MSCAN Wake-Up	I bit	CRIER (WUPIE)	$D0
$FFCA-$FFCF	Reserved	I bit		$CA-$CF
$FFC8, $FFC9	MSCAN Errors	I bit	CRIER (RWRNIE, TWRNIE, RERRIE, TERRIE, BOFFIE, OVRIE)	$C8
$FFC6, $FFC7	MSCAN Receive	I bit	CRIER (RXFIE)	$C6
$FFC4, $FFC5	MSCAN Transmit	I bit	CTCR (TXEIE[2:0])	$C4
$FF80–$FFC3	Reserved	I bit		$80–$C3

Funkcje portów MC68HC12BC32

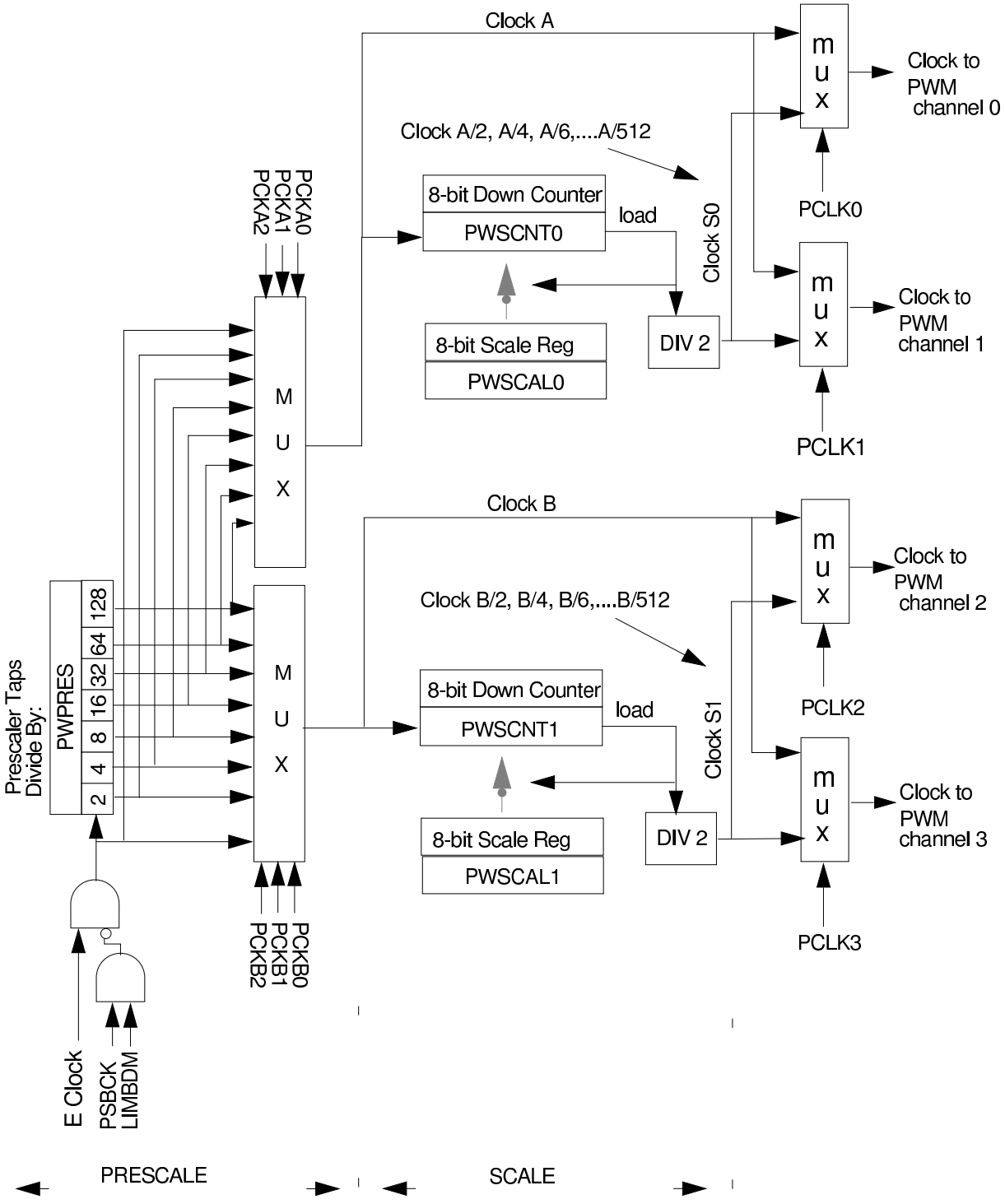
Port Name	Pin Numbers	Data Direction DD Register (Address)	Description
Port A PA[7:0]	46–39	In/Out DDRA ($0002)	Port A and port B pins are used for **address** and **data** in expanded modes. The port data registers are not in the address map during expanded and peripheral mode operation. When in the map, port A and port B can be read or written any time. DDRA and DDRB are not in the address map in expanded or peripheral modes.
Port B PB[7:0]	25–18	In/Out DDRB ($0003)	
Port AD PAD[7:0]	58–51	In	**Analog-to-digital converter** and general-purpose I/O.
Port CAN PCAN[6:2] TxCAN RxCAN	70–76	In/Out DDRCAN ($013F) for PCAN[6:2] TxCAN Out RxCAN In	**CAN Controller** (MSCAN12) subsystem with TxCAN output and RxCAN input and general-purpose I/O on PCAN[6:2].
Port E PE[7:0]	26–29, 35–38	PE[1:0] In PE[7:2] In/Out DDRE ($0009)	**Mode selection**, **bus control** signals and **interrupt service request** signals; or general-purpose I/O.
Port P PP[7:0]	79, 80, 1–6	In/Out DDRP ($0057)	General-purpose I/O. PP[3:0] are use with the **pulse-width modulator** when enabled.
Port S PS[7:0]	68–61	In/Out DDRS ($00D7)	**Serial communications interface** and **serial peripheral interface** subsystems and general-purpose I/O.
Port T PT[7:0]	16–12, 9–7	In/Out DDRT ($00AF)	General-purpose I/O when not enabled for input capture and output compare in the **timer** and **pulse accumulator** subsystem.

Modulacja wypełnienia impulsów (PWM)

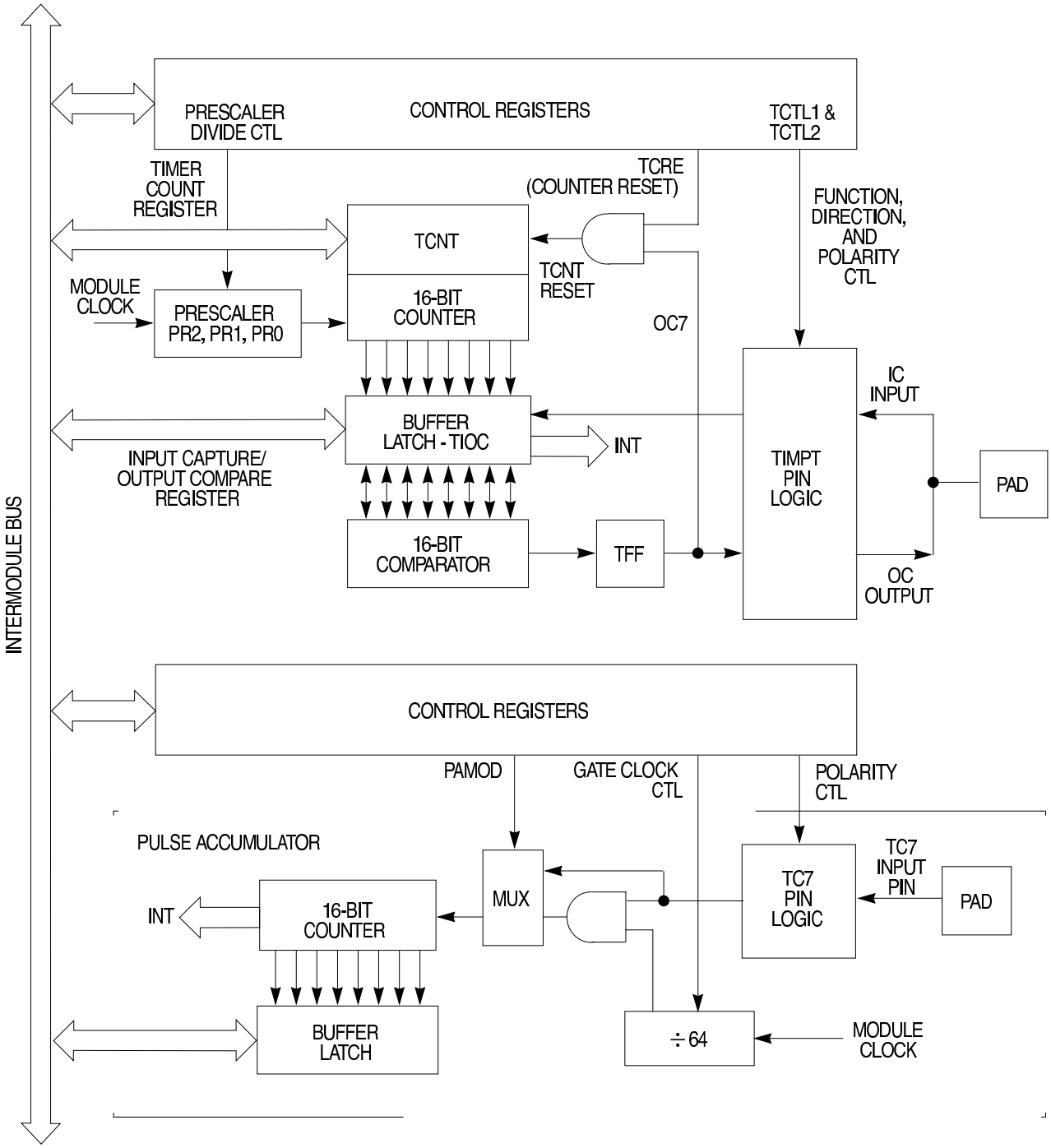




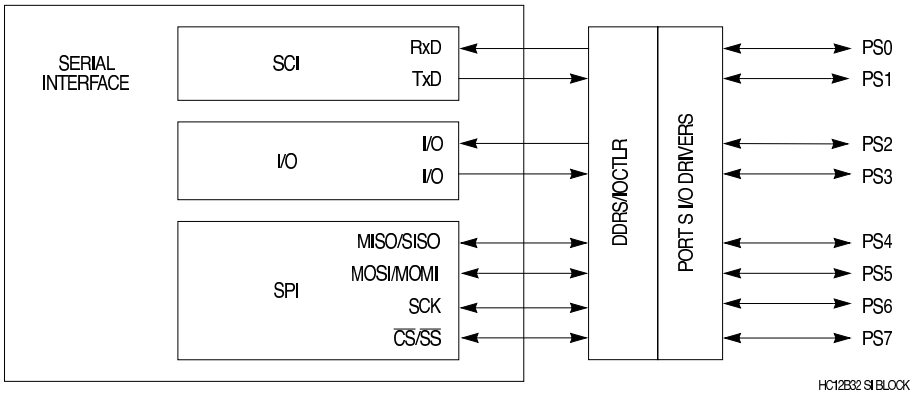
Blok PWM w MC68HC12BC32

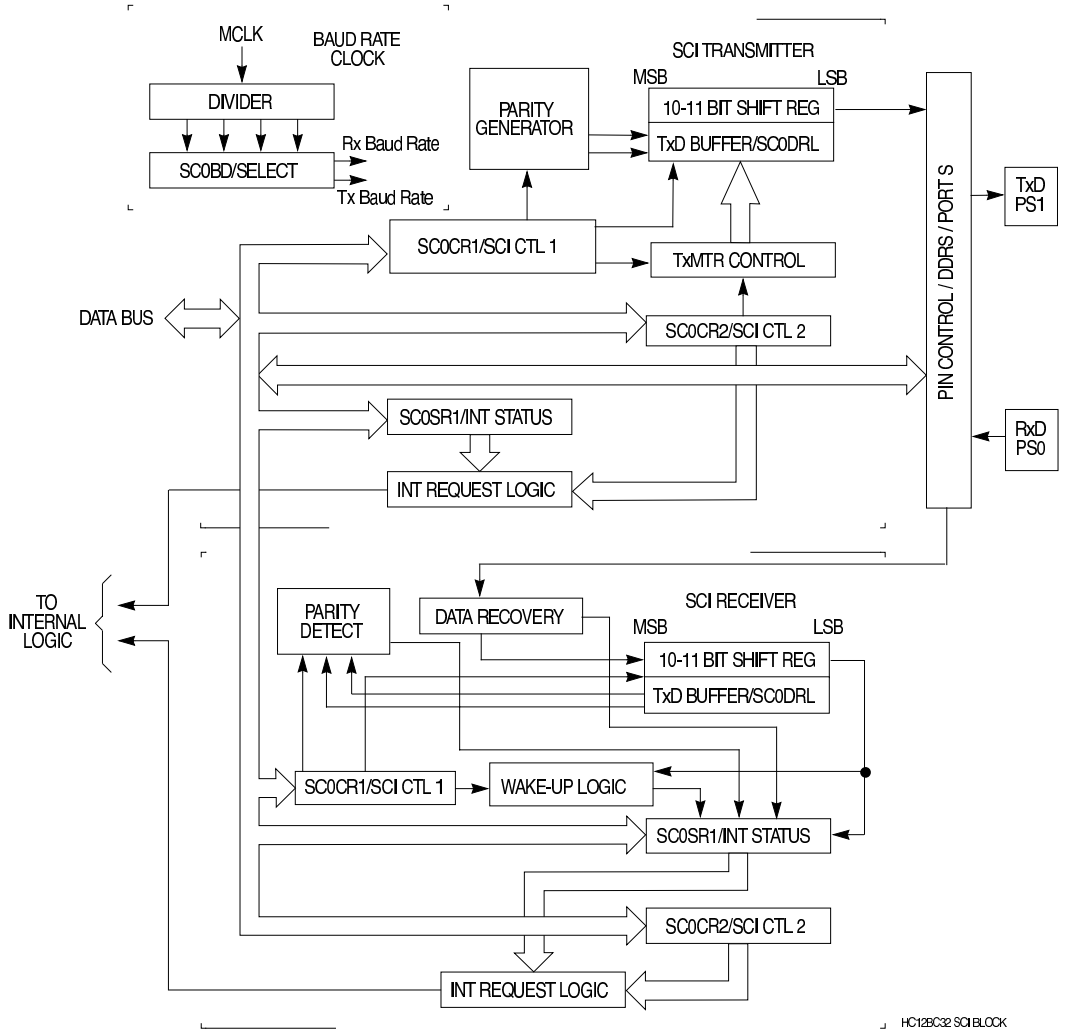


IC, OC i PA w MC68HC12BC32

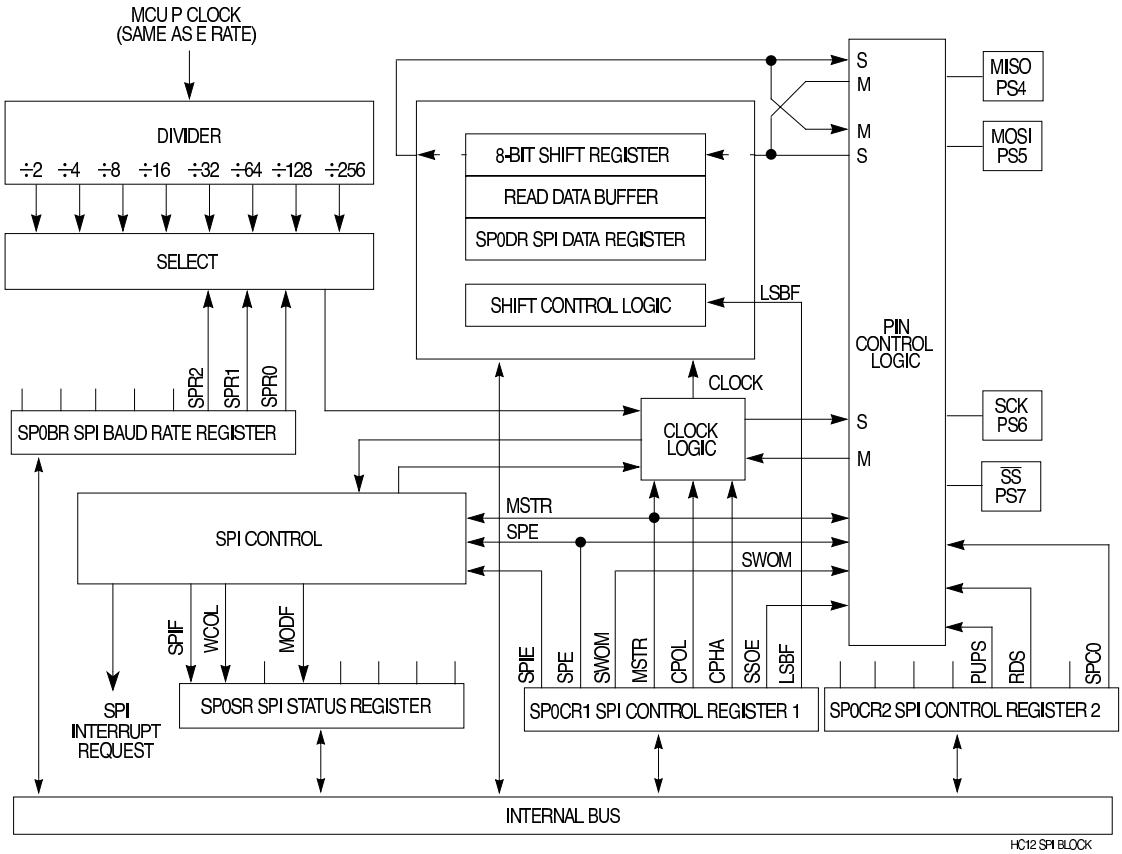


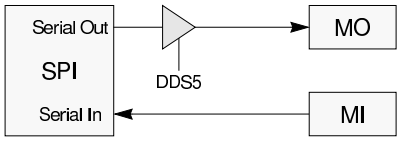
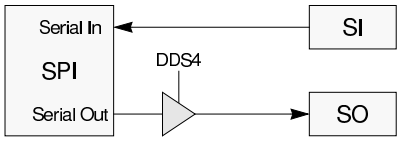
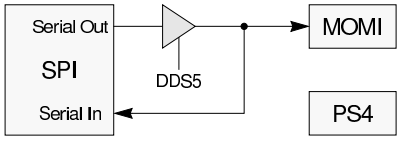
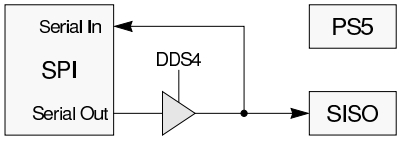
Interfejsy szeregowe - SCI



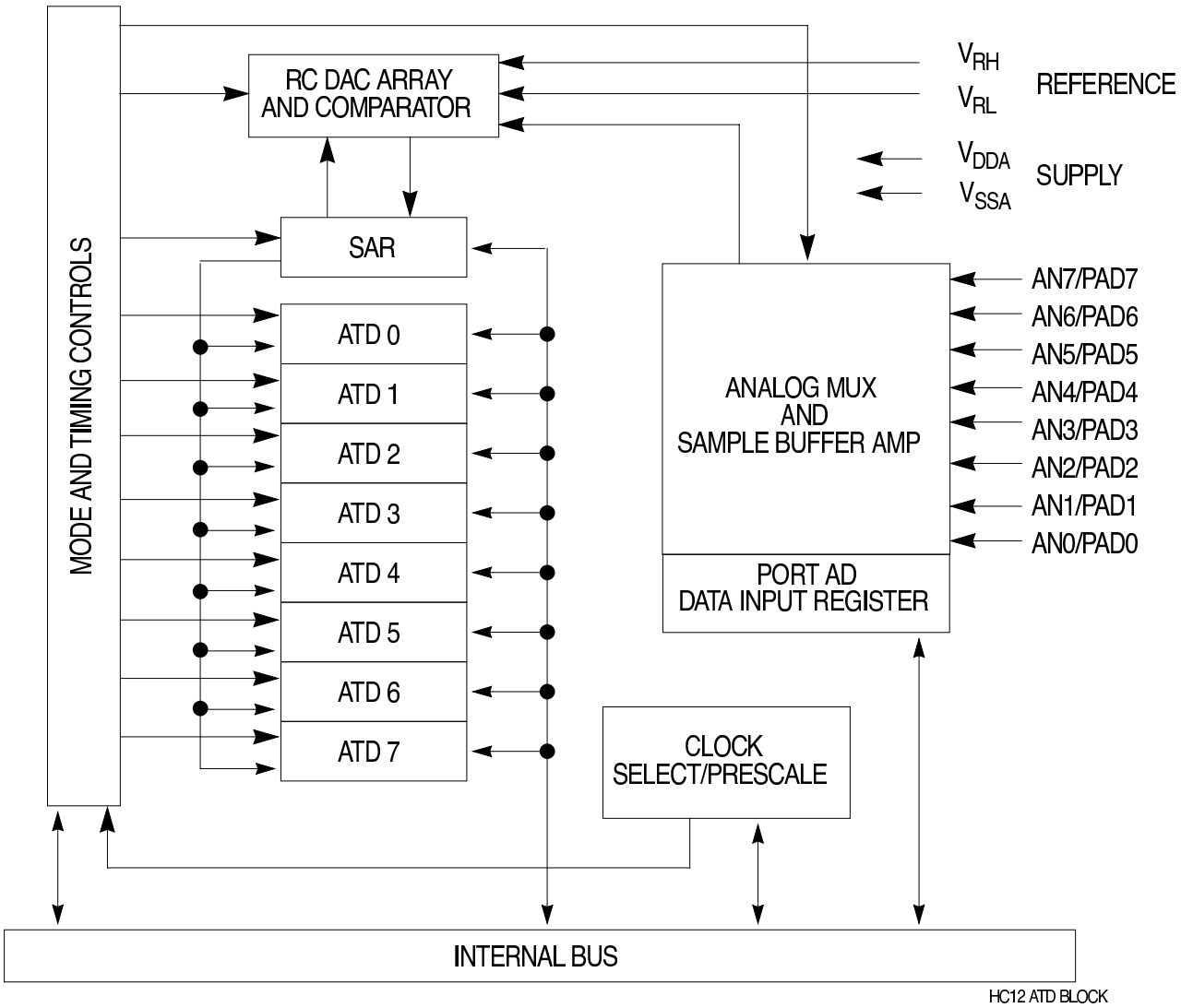


Interfejsy szeregowe - SPI



When SPE=1	Master Mode MSTR=1	Slave Mode MSTR=0
Normal Mode SPC0=0	 SWOM enables open drain output.	 SWOM enables open drain output.
Bidirectional Mode SPC0=1	 SWOM enables open drain output. PS4 becomes GPIO.	 SWOM enables open drain output. PS5 becomes GPIO.

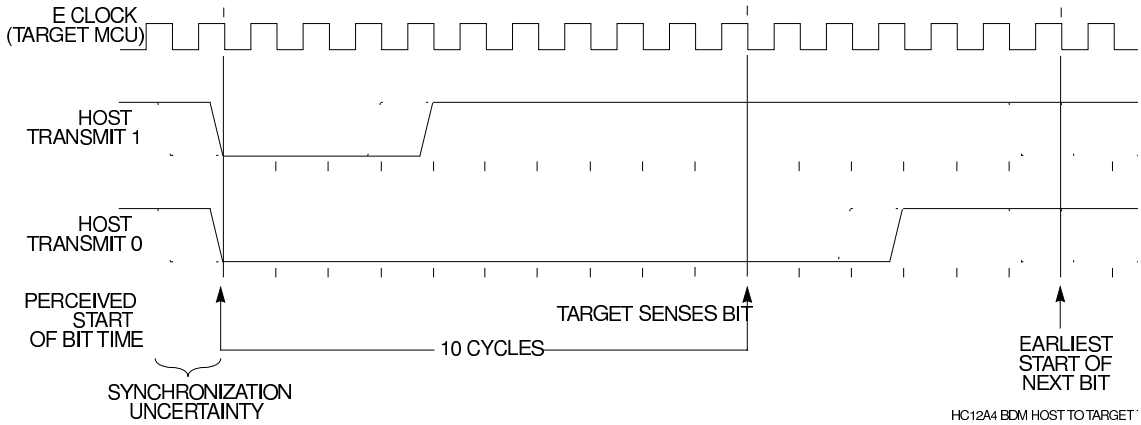
Przetwornik A/C w MC68HC12BC32



Sterowanie przetwornikiem A/C

S8CM	CD	CC	CB	CA	Channel Signal	Result in ADRx if MULT = 1
0	0	0	0	0	AN0	ADR0
			0	1	AN1	ADR1
			1	0	AN2	ADRx
			1	1	AN3	ADR3
0	0	1	0	0	AN4	ADR0
			0	1	AN5	ADR1
			1	0	AN6	ADR2
			1	1	AN7	ADR3
0	1	0	0	0	Reserved	ADR0
			0	1	Reserved	ADR1
			1	0	Reserved	ADR2
			1	1	Reserved	ADR3
0	1	1	0	0	V_{RH}	ADR0
			0	1	V_{RL}	ADR1
			1	0	$(V_{RH} + V_{RL})/2$	ADR2
			1	1	TEST/Reserved	ADR3
1	0	0	0	0	AN0	ADR0
		0	0	1	AN1	ADR1
		0	1	0	AN2	ADR2
		0	1	1	AN3	ADR3
		1	0	0	AN4	ADR4
		1	0	1	AN5	ADR5
		1	1	0	AN6	ADR6
		1	1	1	AN7	ADR7
1	1	0	0	0	Reserved	ADR0
		0	0	1	Reserved	ADR1
		0	1	0	Reserved	ADR2
		0	1	1	Reserved	ADR3
		1	0	0	V_{RH}	ADR4
		1	0	1	V_{RL}	ADR5
		1	1	0	$(V_{RH} + V_{RL})/2$	ADR6
		1	1	1	TEST/Reserved	ADR7

Układ BDM w MC68HC12BC32



Command	Opcode (Hex)	Data	Description
BACKGROUND	90	None	Enter background mode (if firmware enabled).
READ_BD_BYTE	E4	16-bit address 16-bit data out	Read from memory with BDM in map (may steal cycles if external access) data for odd address on low byte, data for even address on high byte.
STATUS[1]	E4	FF01, 0000 0000 (out)	READ_BD_BYTE $FF01. Running user code (BGND instruction is not allowed).
		FF01, 1000 0000 (out)	READ_BD_BYTE $FF01. BGND instruction is allowed.
		FF01, 1100 0000 (out)	READ_BD_BYTE $FF01. Background mode active (waiting for single wire serial command).
READ_BD_WORD	EC	16-bit address 16-bit data out	Read from memory with BDM in map (may steal cycles if external access) must be aligned access.
READ_BYTE	E0	16-bit address 16-bit data out	Read from memory with BDM out of map (may steal cycles if external access) data for odd address on low byte, data for even address on high byte.
READ_WORD	E8	16-bit address 16-bit data out	Read from memory with BDM out of map (may steal cycles if external access) must be aligned access.
WRITE_BD_BYTE	C4	16-bit address 16-bit data in	Write to memory with BDM in map (may steal cycles if external access) data for odd address on low byte, data for even address on high byte.
ENABLE_FIRMWARE[2]	C4	FF01, 1xxx xxxx(in)	Write byte $FF01, set the ENBDM bit. This allows execution of commands which are implemented in firmware. Typically, read STATUS, OR in the MSB, write the result back to STATUS.
WRITE_BD_WORD	CC	16-bit address 16-bit data in	Write to memory with BDM in map (may steal cycles if external access) must be aligned access.
WRITE_BYTE	C0	16-bit address 16-bit data in	Write to memory with BDM out of map (may steal cycles if external access) data for odd address on low byte, data for even address on high byte.
WRITE_WORD	C8	16-bit address 16-bit data in	Write to memory with BDM out of map (may steal cycles if external access) must be aligned access.

NOTES:
1. STATUS command is a specific case of the READ_BD_BYTE command.
2. ENABLE_FIRMWARE is a specific case of the WRITE_BD_BYTE command.

Komendy BDM (cd.)

Command	Opcode (Hex)	Data	Description
READ_NEXT	62	16-bit data out	X = X + 2; Read next word pointed-to by X
READ_PC	63	16-bit data out	Read program counter
READ_D	64	16-bit data out	Read D accumulator
READ_X	65	16-bit data out	Read X index register
READ_Y	66	16-bit data out	Read Y index register
READ_SP	67	16-bit data out	Read stack pointer
WRITE_NEXT	42	16-bit data in	X = X + 2; Write next word pointed-to by X
WRITE_PC	43	16-bit data in	Write program counter
WRITE_D	44	16-bit data in	Write D accumulator
WRITE_X	45	16-bit data in	Write X index register
WRITE_Y	46	16-bit data in	Write Y index register
WRITE_SP	47	16-bit data in	Write stack pointer
GO	08	None	Go to user program
TRACE1	10	None	Execute one user instruction then return to BDM
TAGGO	18	None	Enable tagging and go to user program