MICO Dual Serial Interface for pdp8/e/f/m

Datum : 2006-04-02

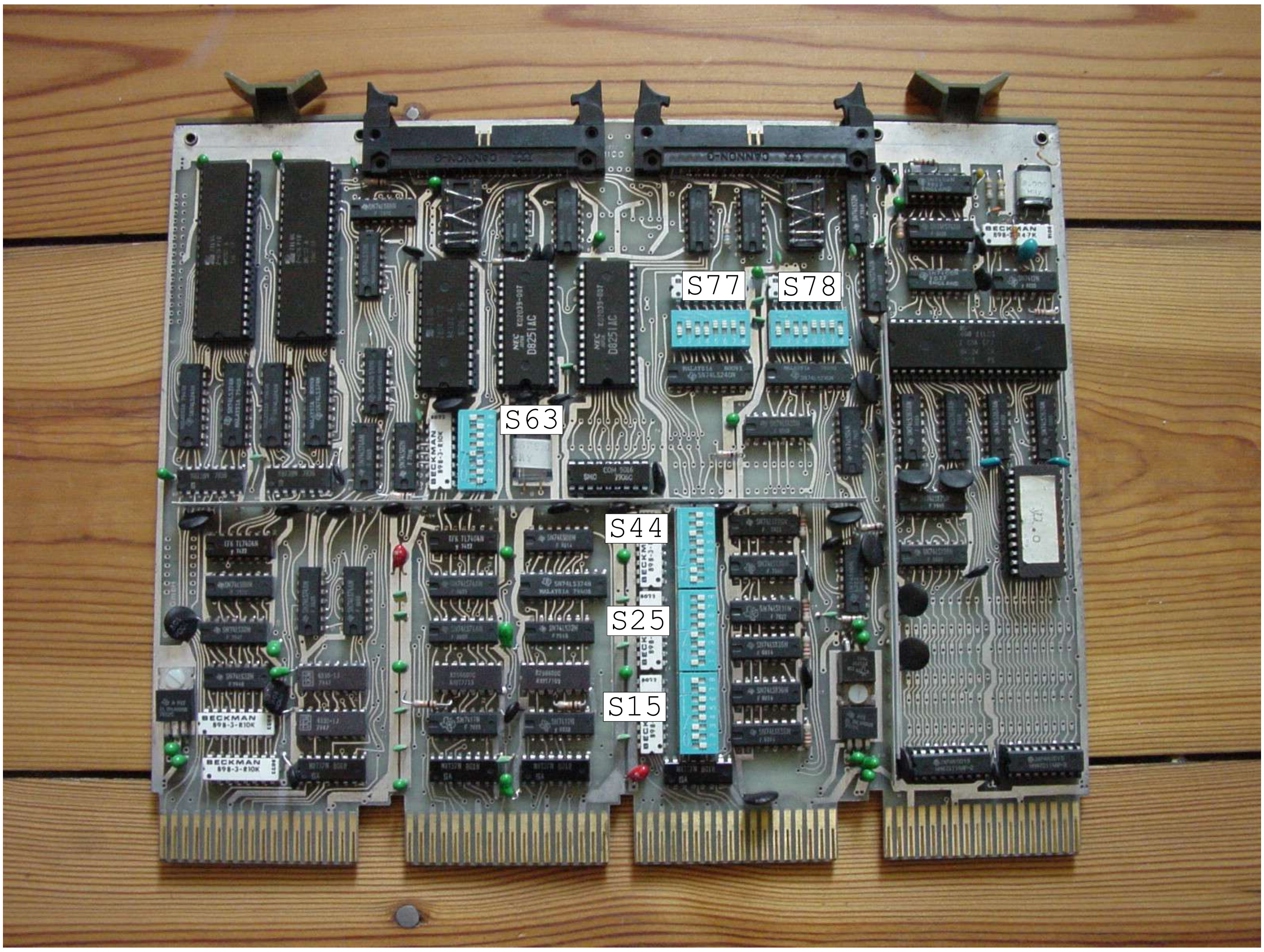

Figure 1

Abstract:

This document describes the dual serial interface module manufactured by MICO GmbH, Freiburg, Germany for pdp8 OMNIBUS type machines like pdp8/e/f/m and /a.

Lacking any description of the module I had to explore the pcb and to disassemble the firmware to get the required information.

I definitely will **not** give any warranties for correctnes or usability of this document.

2006-04-02 Gerold Pauler

Description:

The module can be divided in two regions made out of four building blocks. The two regions are the OMNIBUS interface and the IO area.

mico dual serial interface module

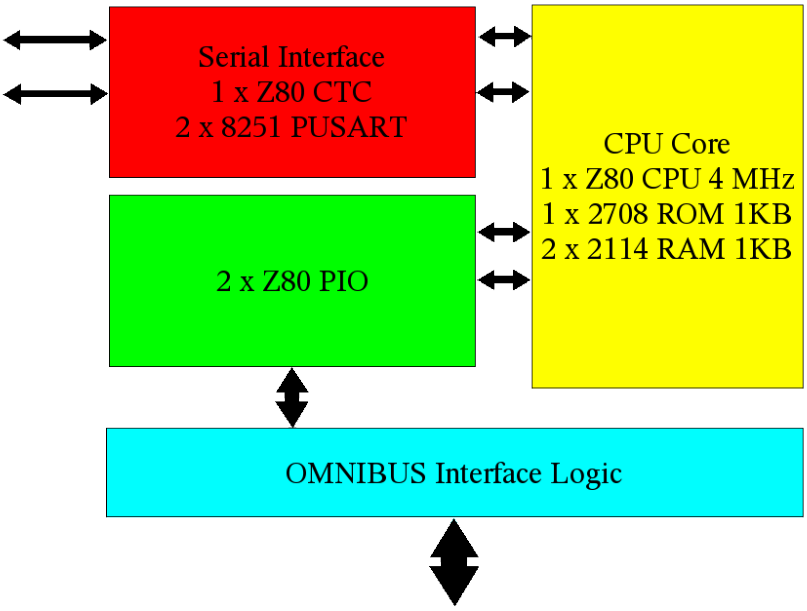

Figure 2

OMNIBUS Interface:

The circuits interfacing the OMNIBUS consist of bus tranceivers (N8T37 for MDx_L and control signals and N8T38 for DATAx_L signals), four banks of device code select switch registers, comparators and the necessary glue logic in 6330 and 29660 bipolar ROMs.

The four device code select switch registers are implemented by three octal switches.

Device	**Receive right hand interface**					
Switch	S15/1	S15/2	S15/3	S15/4	S15/5	S15/6
Device Code Bit	MD3	MD4	MD5	MD6	MD7	MD8

Device	**Transmit right hand interface**					
Switch	S15/7	S15/8	S25/1	S25/2	S25/3	S25/4
Device Code Bit	MD3	MD4	MD5	MD6	MD7	MD8

Device	**Receive left hand interface**					
Switch	S25/5	S25/6	S25/7	S25/8	S44/1	S44/2
Device Code Bit	MD3	MD4	MD5	MD6	MD7	MD8

Device	**Transmit left hand interface**					
Switch	S44/3	S44/4	S44/5	S44/6	S44/7	S44/8
Device Code Bit	MD3	MD4	MD5	MD6	MD7	MD8

Shown in the following table are some common device codes (bits MD3 - MD8 only) for receive and transmit:

Device	Receive	Transmit
Console	03	04
2nd KL8/E	30	31
3rd KL8/E	32	33
4th KL8/E	34	35
5th KL8/E	36	37

IO Area:

This region can be further divided into the CPU core (Z80, up to 4 KByte RAM and 1 KByte ROM), the parallel interface (PIOs) to the OMNIBUS and the serial interface (CTC and SIOs).

CPU Core:

The CPU core consists of a Z80 microprocessor running at 4 MHz, 1 KByte ROM (2708) and 1 KByte static RAM (2 x 2114 - 1K x 4) expandable up to 4 KByte.

Memory Map:

Address range	Type	Description
0000 Hex 03FF Hex	ROM	Interrupt vector table at 0070 Hex Firmware starts at 00A0 Hex 01E4 Hex - 03FF Hex not used
1000 Hex 1400 Hex	RAM	1000 Hex - 1010 Hex stack 1030, 1031 and 1032 Hex scratchpad

The firmware sets the CPU into interrupt mode 2.

The Z80 like the 8080 uses a separate address space to select IO devices. The location of the IO devices on the mico dual serial interface module are shown in the following IO map.

IO Map:		
Address	Device	Description
020 Hex	PIO_1	Port A data register
021 Hex	PIO_1	Port B data register
022 Hex	PIO_1	Port A control register
023 Hex	PIO_1	Port B control register
024 Hex	PIO_0	Port A data register
025 Hex	PIO_0	Port B data register
026 Hex	PIO_0	Port A control register
027 Hex	PIO_0	Port B control register
028 Hex	SWITCH	mode for PUSART_0
02C Hex	PUSART_0	data register
02D Hex	PUSART_0	command/status register
030 Hex	PUSART_1	data register
031 Hex	PUSART_1	command/status register
034 Hex	SWITCH	mode for PUSART_1
038 Hex	CTC	channel 0
039 Hex	CTC	channel 1
03A Hex	CTC	channel 2
03B Hex	CTC	channel 3
03C Hex	???	not known yet

Parallel Interface:

Communication between the CPU core and the OMNIBUS is implemented by two Z80 PIOs with port A of the PIOs running in input only mode and port B running in output only mode.

Serial Interface:

The serial interfaces consist of a Z80 CTC clock timer counter, two 8251 PUSARTs, a COM5016 dual baud rate generator, a switch register bank to select the baud rate for every interface, two switch register banks for the operating modes of the PUSARTs and the necessary RS232 level converters.

The baud rate is selected by the PUSART divisor (default 16x but selectable by the PUSART mode switches) and the divisor selection switches for the baud rate generator.

The following table shows the available selections, assuming the default PUSART divisor (16x) is selected.

Baudrate	Interface	
	rigth	left
	S63/1-4	S63/5-8
19200	0000	0000
9600	0001	0001
4800	0010	010
3600	0011	011
2400	0100	0100
2000	0101	0101
1800	0110	0110
1200	0111	0111
600	1000	1000
300	1001	1001
200	1010	1010
150	1011	1011
134.5	1100	1100
110	1101	1101

Baudrate	Interface	
75	1110	1110
50	1111	1111

Serial communication modes can be set via switches S77 for the left hand interface and S78 for the right hand interface.

The firmware checks the switch settings for illegal mode combinations and disables them by setting the default mode of 8 data bits, no parity, 2 stop bits, divisor = 16x.

Mode Switch	Description				
1 2	0 0	0 1	1 0	1 1	number of stop bits
	invalid	1	1.5	2	
3	0	1			parity
	odd	even			
4	0	1			parity
	disable	enable			
5 6	0 0	0 1	1 0	1 1	number of data bits
	5	6	7	8	
7 8	0 0	0 1	1 0	1 1	divisor
	sync.	1x	16x	64x	sync. not supported

Serial Ports Pinout:

The pinout of the left hand port is compatible to the pinout of an M8650 (KL8E) module. The pinout of the right hand port lacks the brigded connections for pin M, H and pin E.

Commented Firmware Disassembly:

```
;
; mico.src
;
; disassembly listing of mico dual serial interface for pdp8/e/f/m/a
;
; comments by Gerold Pauler 2006-03-31 - 2006-04-02
;
; Memory Map
;
;           0000H        2708 ROM: Reset Vector
;           0070H            Interrupt Vector table for IM 2
;           007FH
;           00A0H            POWER ON reset
;           013DH            service loop
;           01E3H            end of firmware
;           01E4H            not used
;           03FFH            end of 2708 ROM
;
:           1000H        1 K RAM (2 x 2114): Stack
;           1010H            top of stack
;           1030H            WOM (???)
;           1031H            WOM (???)
;           1032H            WOM (???)
;
; IO Map
;
;           020H         PIO_1 Port A data register
;           021H         PIO_1 Port B data register
;           022H         PIO_1 Port A control register
;           023H         PIO_1 Port B control register
;
;           024H         PIO_0 Port A data register
;           025H         PIO_0 Port B data register
;           026H         PIO_0 Port A control register
;           027H         PIO_0 Port B control register
;
;           028H         mode switch for PUSART_0
;
;           02CH         PUSART_0 data register
;           02DH         PUSART_0 command/status register
;
;           030H         PUSART_1 data register
;           031H         PUSART_1 command/status register
;
;           034H         mode switch for PUSART_1
;
;           038H         CTC channel 0
;           039H         CTC channel 1
;           03AH         CTC channel 2
;           03BH         CTC channel 3
;
;           03CH         ???
;
            ORG          00000H
            JP           000A0H      ; 0000
;
            ORG          00008H
            JP           000A0H      ; 0008
;
            ORG          00010H
            JP           000A0H      ; 0010
;
            ORG          00018H
            JP           000A0H      ; 0018
```

```
;
            ORG         00020H
            JP          000A0H      ; 0020
;
            ORG         00028H
            JP          000A0H      ; 0028
;
            ORG         00030H
            JP          000A0H      ; 0030
;
            ORG         00038H
            JP          000A0H      ; 0038
;
            ORG         00066H
            JP          000A0H      ; 0066
;
; interrupt vector table
;
            ORG         00070H
            DW          001BDH      ; 0070: interrupt vector CTC channel 0
            DW          001B9H      ; 0072: interrupt vector CTC channel 1
            DW          001D2H      ; 0074: interrupt vector CTC channel 2
            DW          001C1H      ; 0076: interrupt vector CTC channel 3
            DW          001B1H      ; 0078: interrupt vector PIO_1 input port A
            DW          001A9H      ; 007A: interrupt vector PIO_1 output port B
            DW          001B5H      ; 007C: interrupt vector PIO_0 input port A
            DW          001ADH      ; 007E: interrupt vector PIO_0 output port B
;
; POWER-ON RESET
;
            ORG         000A0H
            DI                      ; 00A0: disable interrupts
            LD          SP,01010H   ; 00A1: begin of stack is 1010 H
            IM          2           ; 00A4: set interrupt mode 2
            LD          A,000H      ; 00A6: interrupt vector table starts at 0
            LD          I,A         ; 00A8
;
            IN          A,(03CH)    ; 00AA
;
; fill interrupt vector registers of PIO ports
;
            LD          A,078H      ; 00AC: PIO_1 port A
            OUT         (022H),A    ; 00AE: PIO_1 Port A control register
            LD          A,07AH      ; 00B0: PIO_1 port B
            OUT         (023H),A    ; 00B2: PIO_1 Port B control register
            LD          A,07CH      ; 00B4: PIO_0 port A
            OUT         (026H),A    ; 00B6: PIO_0 Port A control register
            LD          A,07EH      ; 00B8: PIO_0 port B
            OUT         (027H),A    ; 00BA: PIO_0 Port B control register
;
; port A of PIOs for Input
;
            LD          A,04FH              ; 00BC
            OUT         (022H),A    ; 00BE: PIO_1 Port A control register
            OUT         (026H),A    ; 00C0: PIO_0 Port A control register
;
; port B of PIOs for Output
;
            LD          A,00FH              ; 00C2
            OUT         (023H),A    ; 00C4: PIO_1 Port B control register
            OUT         (027H),A    ; 00C6: PIO_0 Port B control register
;
; enable interrupts for all PIO ports
;
            LD          A,087H              ; 00C8
            OUT         (022H),A    ; 00CA: PIO_1 Port A control register
            OUT         (026H),A    ; 00CC: PIO_0 Port A control register
            OUT          (023H),A    ; 00CE: PIO_1 Port B control register
            OUT          (027H),A    ; 00D0: PIO_0 Port B control register
;
; read stale data from PIO ports A
;
            IN           A,(020H)    ; 00D2: PIO_1 Port A data register
            IN           A,(024H)    ; 00D4: PIO_0 Port A data register
;
; setup all 4 channels of CTC
; 043H = 0100 0011B
;        |||| |||+-- control
;        |||| ||+--- software reset
;        |||| |+---- no constant
;        |||| +----- start after constant
;        |||+------- trailing edge
;        ||+-------- divisor = 16
;        |+--------- counter mode
;        +---------- interrupt disabled
;
            LD           B,004H      ; 00D6: count := 4
            LD           C,038H      ; 00D8: address of CTC channel 0
            LD           A,043H      ; 00DA: software reset, counter
            OUT          (C),A       ; 00DC
            INC          C           ; 00DE: address of next channel
            DJNZ         000DCH      ; 00DF: all 4 channels done?
;
; reset both PUSARTs
;
            XOR          A           ; 00E1: clear registers
            OUT          (031H),A    ; 00E2: PUSART_1 command
            OUT          (031H),A    ; 00E4: PUSART_1 command
            OUT          (031H),A    ; 00E6: PUSART_1 command
            OUT          (02DH),A    ; 00E8: PUSART_0 command
            OUT          (02DH),A    ; 00EA: PUSART_0 command
            OUT          (02DH),A    ; 00EC: PUSART_0 command
            LD           A,040H      ; 00EE: software reset
            OUT          (031H),A    ; 00F0: PUSART_1 command
            OUT          (02DH),A    ; 00F2: PUSART_0 command
;
;
;
            IN           A,(034H)    ; 00F4: mode switch for PUSART_1
            LD           D,A         ; 00F6
            AND          0C0H        ; 00F7: invalid stop bits?
            JR           Z,00100H    ; 00F9: then take default (0CE H)
            LD           A,D         ; 00FB
            AND          003H        ; 00FC: sync mode?
            JR           NZ,00102H   ; 00FE: then take default (0CE H)
;
; default value 0CEH
; baud rate divisor = 16
; 8 data bits
; no parity
; 2 stop bits
;
; 0CEH = 1100 1110B
;        |||| |||+--- 0   1   0   1
;        |||| ||+---- 0   0   1   1
;        |||| ||     sync 1x 16x 64x baud rate divisor
;        |||| |+----- 0   1   0   1
;        |||| +------ 0   0   1   1
;        ||||         5   6   7   8 data bits
;        |||+-------- no parity
;        ||+--------- odd parity
;        |+---------- 0   1   0   1
;        +----------- 0   0   1   1
;                    inv. 1  1.5  2 stop bits
```

```
;
            LD          D,0CEH      ; 0100: 8N2 divisor 16
;
            LD          A,D         ; 0102
            OUT         (031H),A    ; 0103: PUSART_1 command
;
            IN          A,(028H)    ; 0105: mode switch for PUSART_0
            LD          D,A         ; 0107
            AND         0C0H        ; 0108: invalid stop bits?
            JR          Z,00111H    ; 010A: then take default (0CE H)
            LD          A,D         ; 010C
            AND         003H        ; 010D: sync mode?
            JR          NZ,00113H   ; 010F: then take default (0CE H)
;
            LD          D,0CEH      ; 0111: 8N2 divisor 16
;
            LD          A,D         ; 0113
            OUT         (02DH),A    ; 0114: PUSART_0 command
;
; 037H = 0011 1111B
;        |||| |||+--- enable transmit
;        |||| ||+---- ~DTR LOW
;        |||| |+----- enable receive
;        |||| +------ send break
;        |||+-------- reset error flags
;        ||+--------- ~RTS LOW
;        |+---------- no reset
;        +----------- hunt mode off
;
            LD          A,037H                  ; 0116
            OUT         (031H),A    ; 0118: PUSART_1 command
            OUT         (02DH),A    ; 011A: PUSART_0 command
;
            IN          A,(02CH)    ; 011C: PUSART_0 data
            IN          A,(02CH)    ; 011E: PUSART_0 data
            IN          A,(030H)    ; 0120: PUSART_1 data
            IN          A,(030H)    ; 0122: PUSART_1 data
;
; setup CTC interrupt vector register
;
            LD          C,038H      ; 0124: CTC channel 0
            LD          A,070H      ; 0126: set interrupt vector for CTC
            OUT         (C),A       ; 0128
;
; setup all 4 channels of CTC
; 0D7H = 1101 0111B
;        |||| |||+-- control
;        |||| ||+--- software reset
;        |||| |+---- constant follows
;        |||| +----- start after constant
;        |||+------- leading edge
;        ||+-------- divisor = 16
;        |+--------- counter mode
;        +---------- interrupt enabled
;
            LD          A,0D7H      ; 012A
            LD          D,001H      ; 012C: constant := 1
            LD          B,004H      ; 012E: counter := 4
;
            OUT         (C),A       ; 0130: set mode
            OUT         (C),D       ; 0132: write constant
            INC         C           ; 0134: address of next channel
            DJNZ        00130H      ; 0135: all 4 channels done?
;
            LD          D,03CH      ; 0137: preset bits 2 - 5 (all interrupts serviced)
            LD          HL,00000H   ; 0139: reset H and L
            EI                      ; 013C: enable interrupts
```

```
;
; initialization completed
;


;
; LOOP FOREVER
;           This is an endless loop servicing every device
;           which reset its service bit in register D
;
L4EVER      IN          A,(03CH)    ; 013D
            DI                      ; 013F: disable interrupts
            LD          A,D         ; 0140: save D in memory[1030 H]
            LD          (01030H),A  ; 0141
;
            IN          A,(031H)    ; 0144: PUSART_1 status
            LD          (01031H),A  ; 0146: save PUSART_1 status in memory[1031 H]
            IN          A,(031H)    ; 0149: PUSART_1 status
            AND         038H        ; 014B: errors ?
            JR          Z,0015AH    ; 014D: no then next PUSART_0
            LD          A,037H      ; 014F: error reset and enable
            BIT         6,D         ; 0151: bit 6 in D set?
            JR          NZ,00159H   ; 0153: yes then jump 0159 H
            OUT         (031H),A    ; 0155: PUSART_1 command
            JR          0015AH      ; 0157: next PUSART_0
;
            LD          H,A         ; 0159: flag service error PUSART_1 in H
;
            IN          A,(02DH)    ; 015A: PUSART_0 status
            LD          (01032H),A  ; 015C: save PUSART_0 status in memory[1032 H]
            IN          A,(02DH)    ; 015F: PUSART_0 status
            AND         038H        ; 0161: errors ?
            JR          Z,00170H    ; 0163: no then move PUSART_1 to OMNIBUS
            LD          A,037H      ; 0165: error reset and enable
            BIT         7,D         ; 0167: bit 7 in D set?
            JR          NZ,0016FH   ; 0169: yes then jump 016F H
            OUT         (02DH),A    ; 016B: PUSART_0 command
            JR          00170H      ; 016D: next move PUSART_1 to OMNIBUS
;
            LD          L,A         ; 016F: flag service error PUSART_0 in L
;
            EI                      ; 0170: enable interrupts
;
; copy serial data from PUSARTs to OMNIBUS (PIOs)
;
            LD          A,D         ; 0171: bits 4 or 0 still set?
            AND         011H        ; 0172
            JR          NZ,0017EH   ; 0174: yes then move PUSART_0 to OMNIBUS
            SET         4,D         ; 0176: set bit 4 - mark serviced
            IN          A,(030H)    ; 0178: PUSART_1 data
            SET         0,D         ; 017A: set bit 0 - mark serviced
            OUT         (021H),A    ; 017C: PIO_1 Port B data register
;
            LD          A,D         ; 017E: bits 5 or 1 still set?
            AND         022H        ; 017F
            JR          NZ,0018BH   ; 0181: yes then move OMNIBUS to PUSART_1
            SET         5,D         ; 0183: set bit 5 - mark serviced
            IN          A,(02CH)    ; 0185: PUSART_0 data
            SET         1,D         ; 0187: set bit 1 - mark serviced
            OUT         (025H),A    ; 0189: PIO_0 Port B data register
;
; copy OMNIBUS data (PIOs) to PUSARTs
;
            LD          A,D         ; 018B: bits 6 or 2 still set?
            AND         044H        ; 018C
            JR          NZ,00198H   ; 018E: yes then move OMNIBUS to PUSART_0
            SET         2,D         ; 0190: set bit 2 - mark serviced
            IN          A,(020H)    ; 0192: PIO_1 Port A data register
            OUT         (030H),A    ; 0194: PUSART_1 data
            SET         6,D         ; 0196: set bit 6 - mark serviced
;
            LD          A,D         ; 0198: bits 7 or 3 still set?
            AND         088H        ; 0199
            JP          NZ,L4EVER   ; 019B: yes then loop forever (JP 0013D H)
            SET         3,D         ; 019E: set bit 3 - mark serviced
            IN          A,(024H)    ; 01A0: PIO_0 Port A data register
            OUT         (02CH),A    ; 01A2: PUSART_0 data
            SET         7,D         ; 01A4: set bit 7 - mark serviced
            JP          L4EVER      ; 01A6: loop forever (JP 0013D H)
;
; Interrupt service routines
;           only reset service bit in cpu register D
;           (and reset error flags)
;
; output to OMNIBUS (PIO_1 port B)
;
            RES         0,D         ; 01A9
            JR          001E2H      ; 01AB: interrupt return
;
; output to OMNIBUS (PIO_0 port B)
;
            RES         1,D         ; 01AD
            JR          001E2H      ; 01AF: interrupt return
;
; input from OMNIBUS (PIO_1 port A)
;
            RES         2,D         ; 01B1
            JR          001E2H      ; 01B3: interrupt return
;
; input from OMNIBUS (PIO_0 port A)
;
            RES         3,D         ; 01B5
            JR          001E2H      ; 01B7: interrupt return
;
; CTC channel 1
;
            RES         4,D         ; 01B9
            JR          001E2H      ; 01BB: interrupt return
;
; CTC channel 0
;
            RES         5,D         ; 01BD
            JR          001E2H      ; 01BF: interrupt return
;
; CTC channel 3
;
            EX          AF,AF'      ; 01C1
            XOR         A           ; 01C2: error flaged?
            OR          H           ; 01C3
            JR          NZ,001CAH   ; 01C4: yes then reset errors
            RES         6,D         ; 01C6
            JR          001E1H      ; 01C8: restore interrupt return
;
; reset PUSART_1 errors and error flag H
;
            OUT         (031H),A    ; 01CA: PUSART_1 command
            RES         6,D         ; 01CC
            LD          H,000H      ; 01CE: reset error flag
            JR          001E1H      ; 01D0: restore interrupt return
;
; CTC channel 2
;
            EX          AF,AF'      ; 01D2
            XOR         A           ; 01D3: error flaged?
            OR          L           ; 01D4
            JR          NZ,001DBH   ; 01D5: yes then reset errors
            RES         7,D         ; 01D7
            JR          001E1H      ; 01D9: restore interrupt return
;
; reset PUSART_0 errors and error flag L
;
            OUT         (02DH),A    ; 01DB: PUSART_0 command
            RES         7,D         ; 01DD
            LD          L,000H      ; 01DF: reset error flag
;
; restore interrupt return
;
            EX          AF,AF'      ; 01E1
;
            EI                      ; 01E2: enable interrupts
            RETI                    ; 01E3
            END
```