

MON51 Board

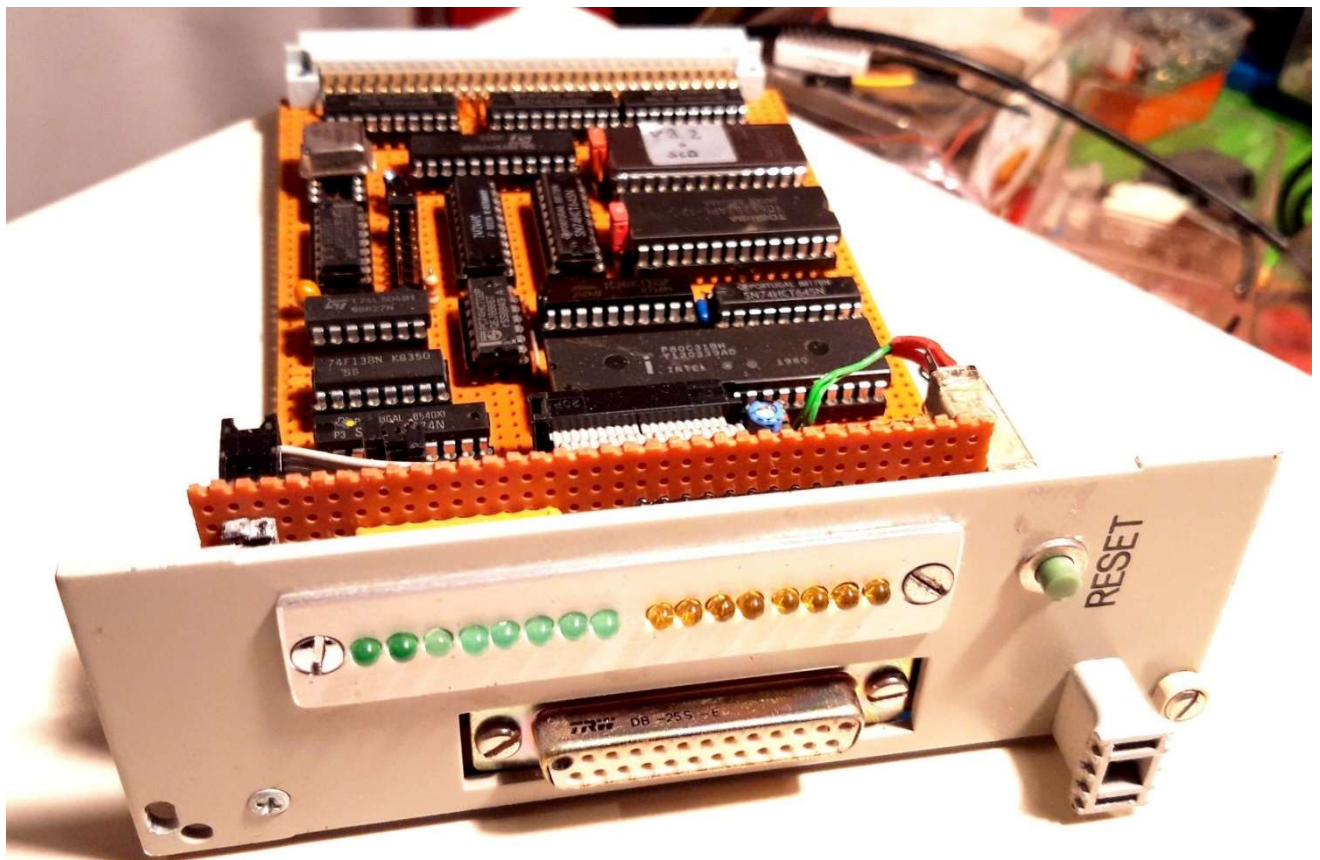
Mit 8031/51 CPU

Angeschlossen und gesteuert über ein AT Bus basiertes
PC-System

02.07.1996 - Randolph Esser

Technical Data

Prozessor:	i8031/51
Clock:	11.059MHz
SDRam:	16/32kByte (4364/ 43256)
EPRom:	16/32kByte (27C128/ 27C256)



AT-Bus-Interface Timing

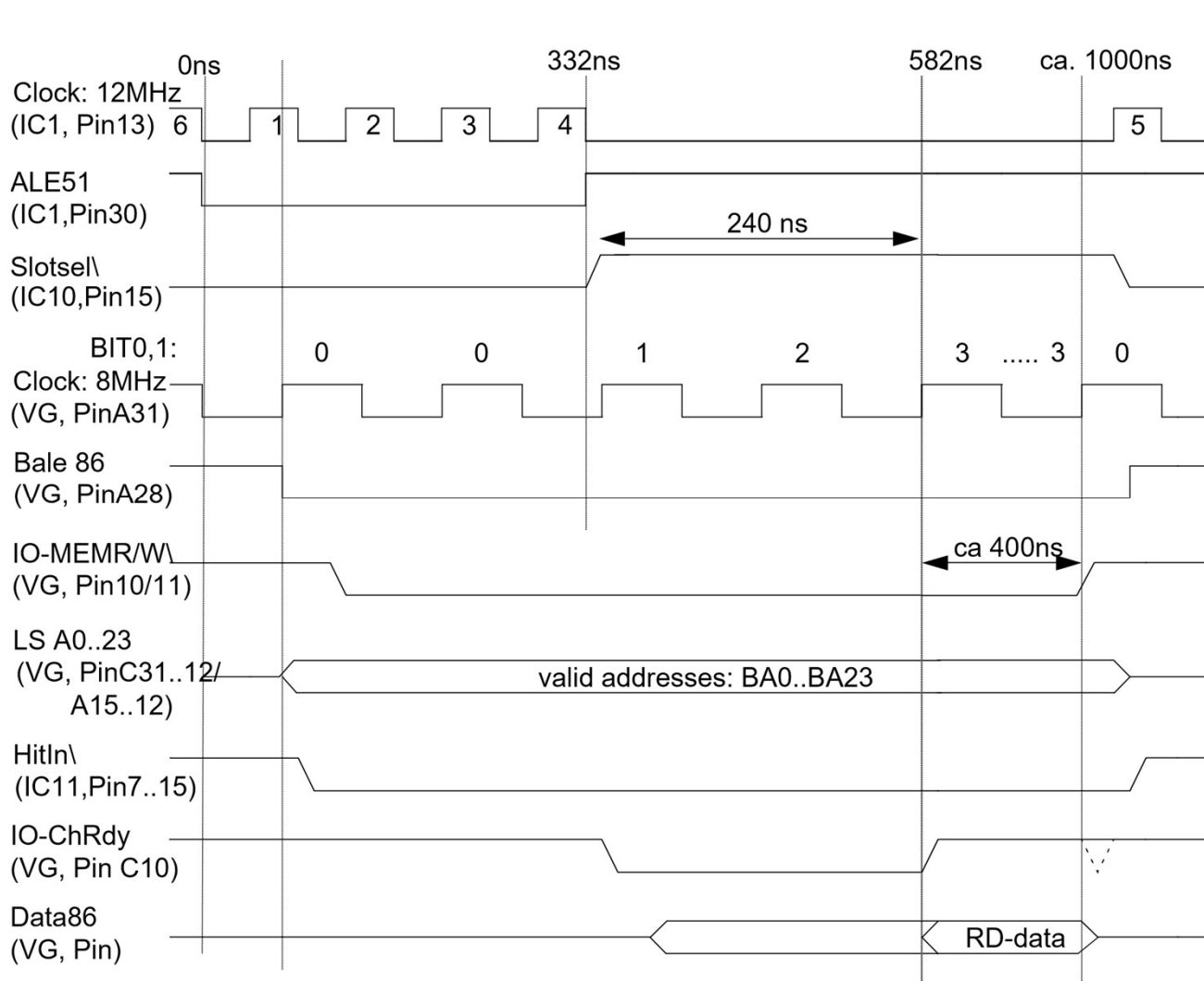
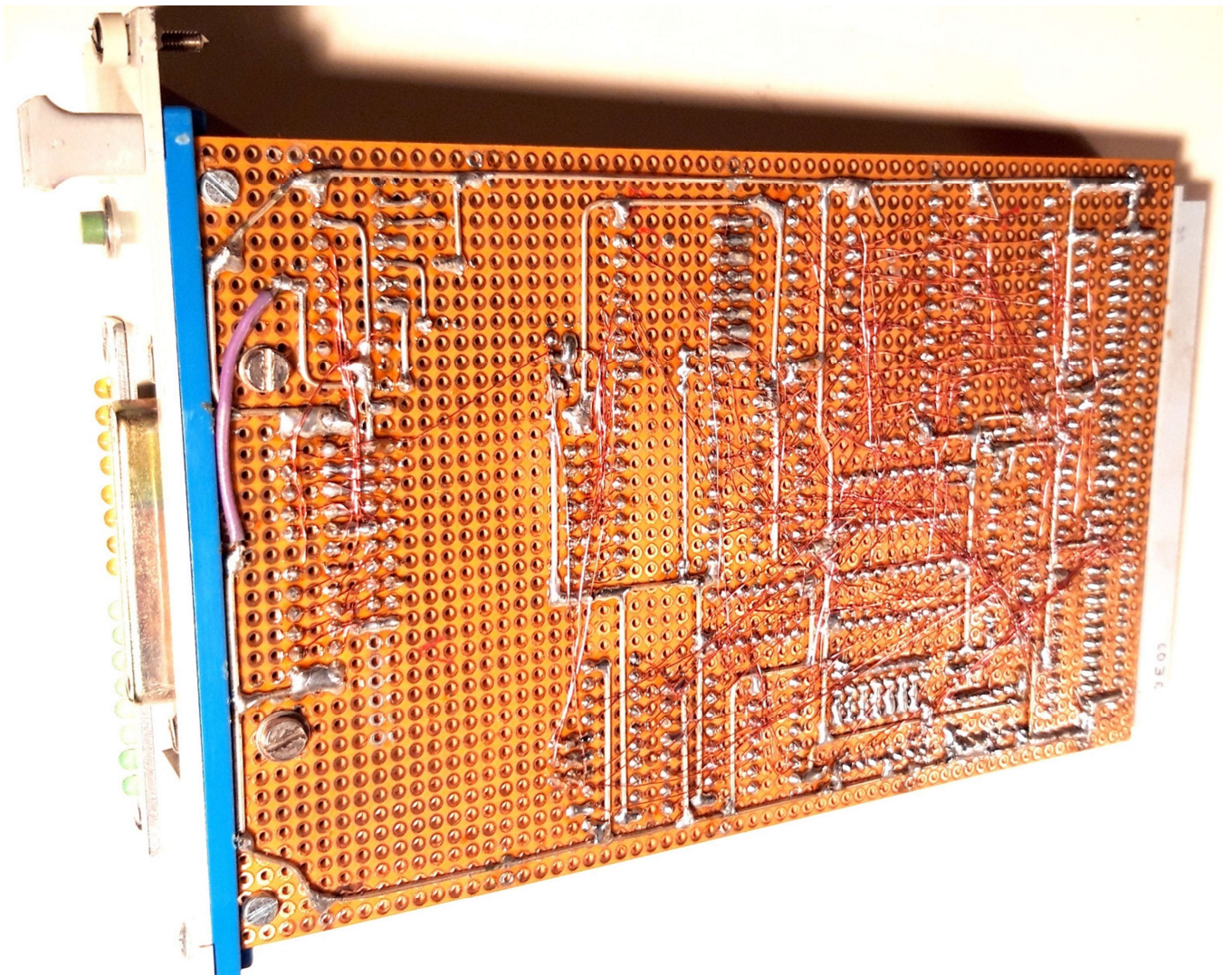
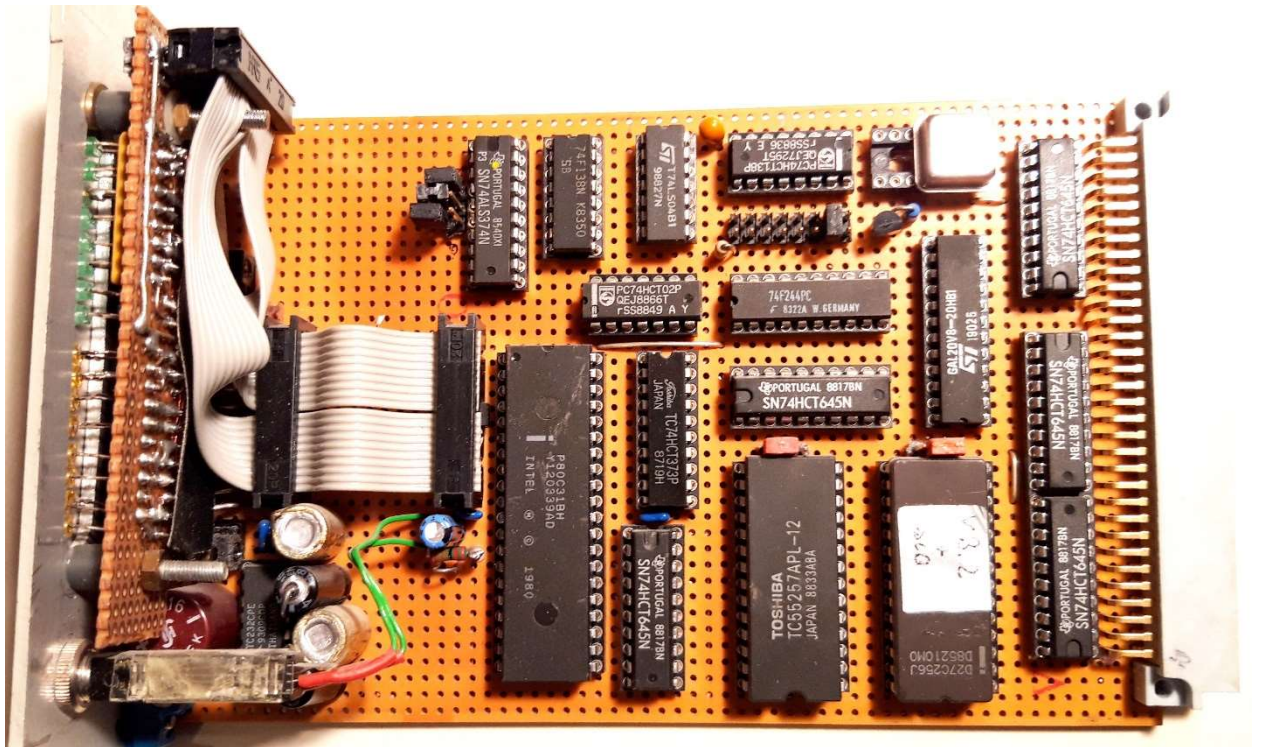
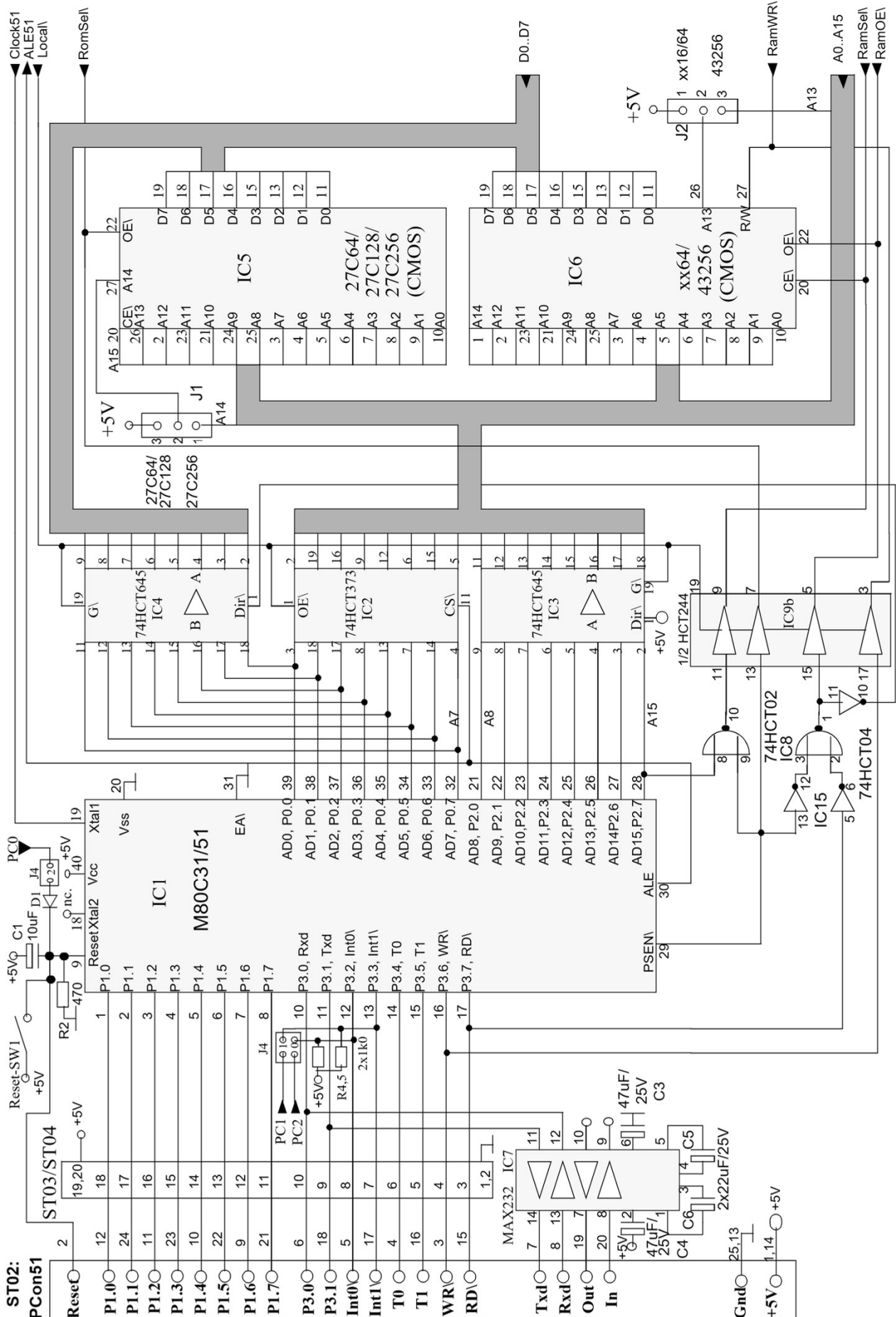


Abb1: M80x1 / AT Bus handshake Timing-Diagramm

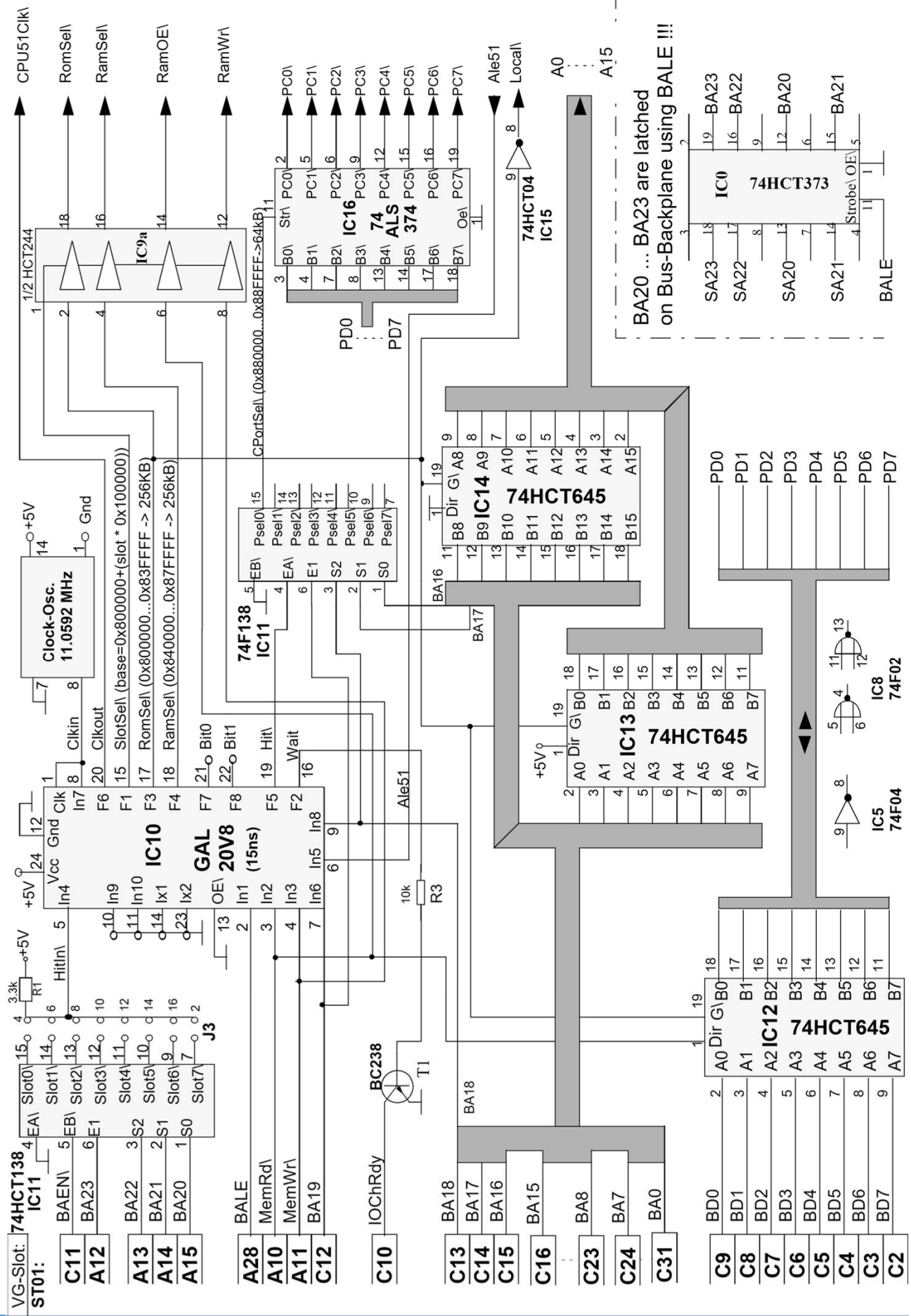
MON51 Top + Bottom View



Schematics: Process-control + Memory/Rom+ Port-Interfaces

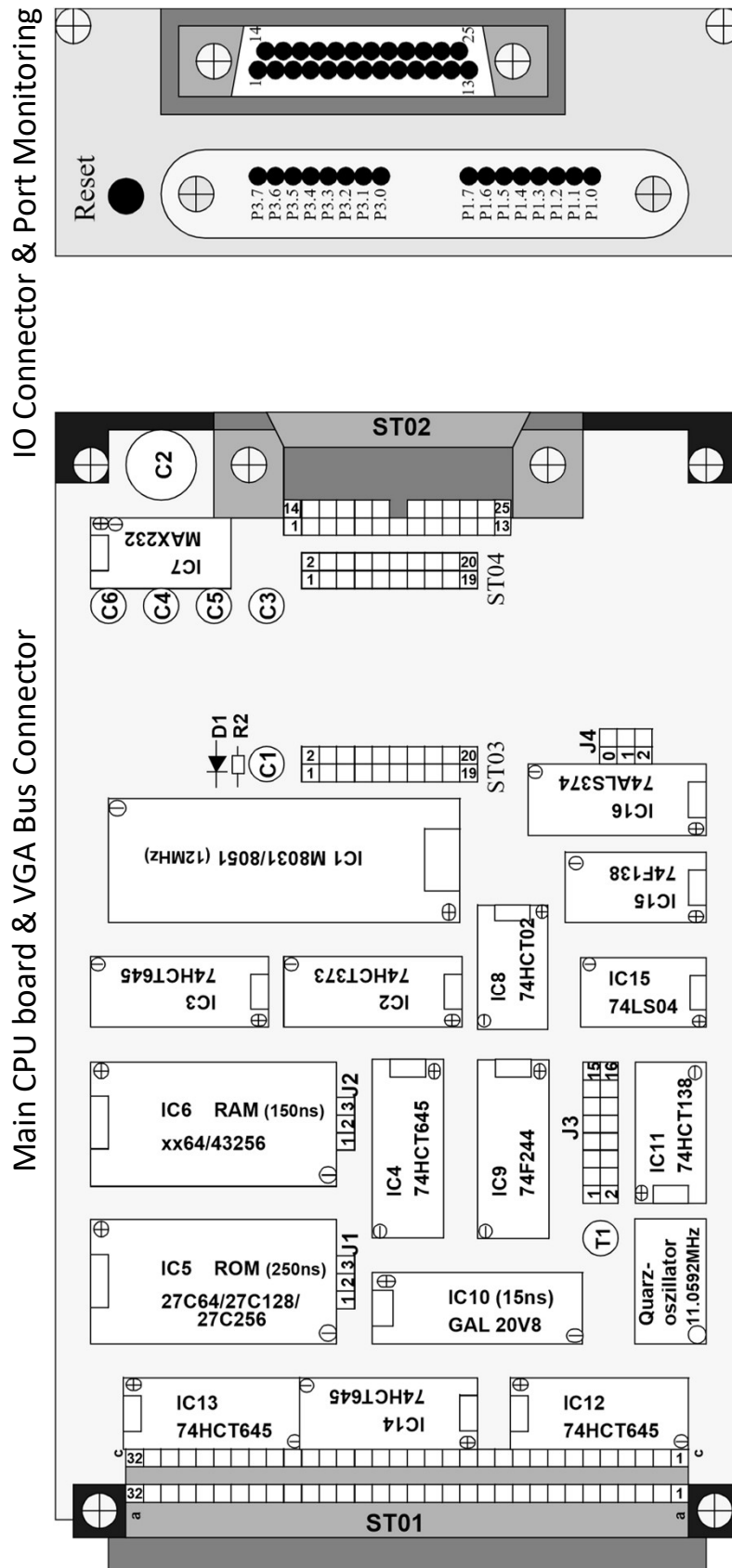


ATBus-VG-Slotcard: Address Decoding + VG96 Bus-Driver

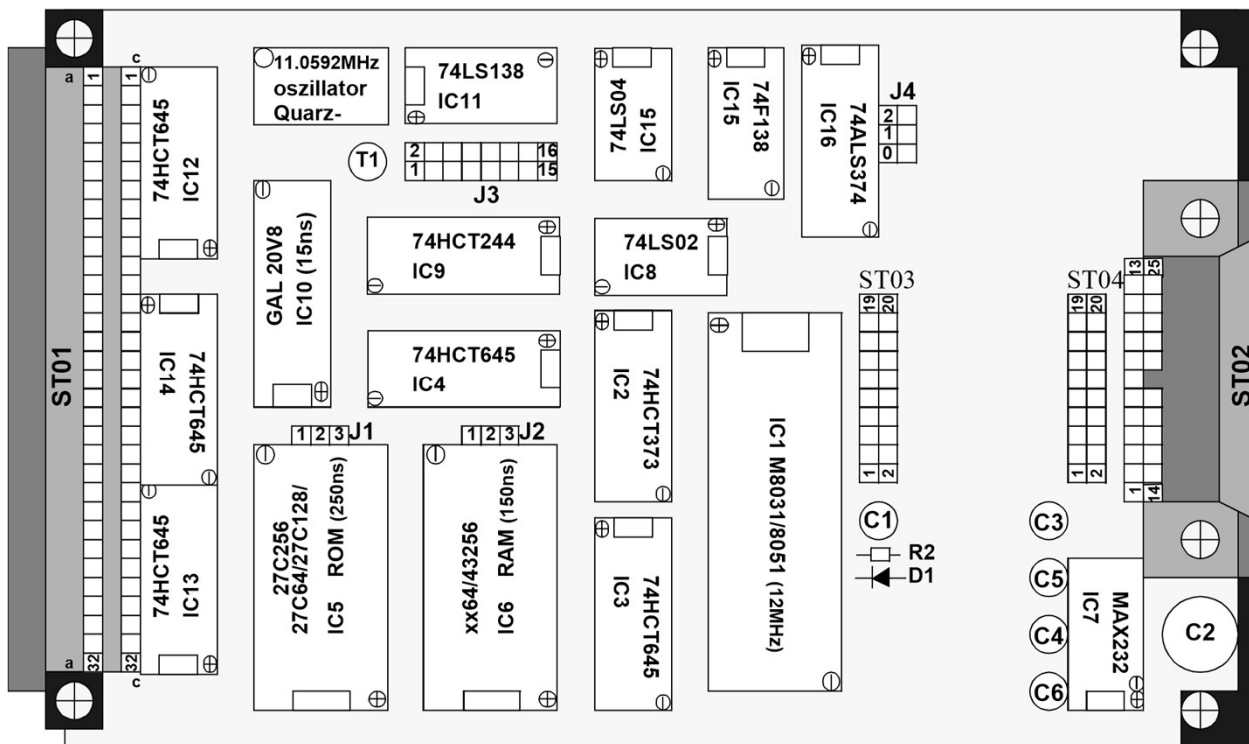


Boardlayout: 19"- Slotcard with VGA connector (male)

Top Side



Boardlayout: 19"- Slotcard with VGA connector
Bottom Side



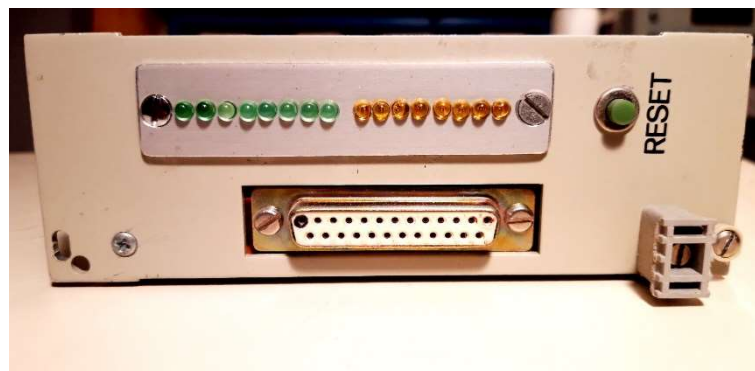
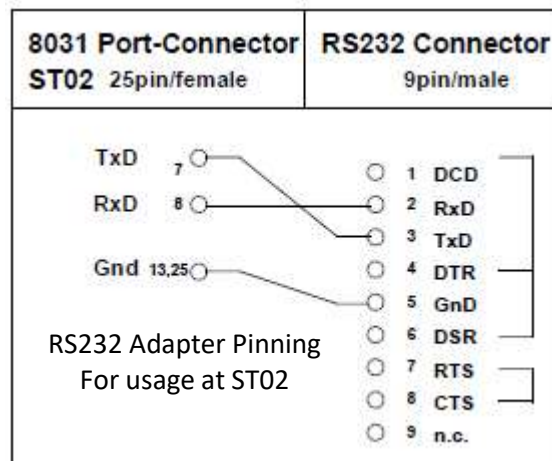
J3: Jumper for Slot Base Address Selection to Pin5 IC10 (GAL)

Slot#	IC1	J3	Slot Adr
0	15	4	0x800000
1	14	6	0x900000
2	13	8	0xA00000
3	12	10	0xB00000
4	11	12	0xC00000
5	10	14	0xD00000
6	9	16	0xE00000
7	7	2	0xF00000

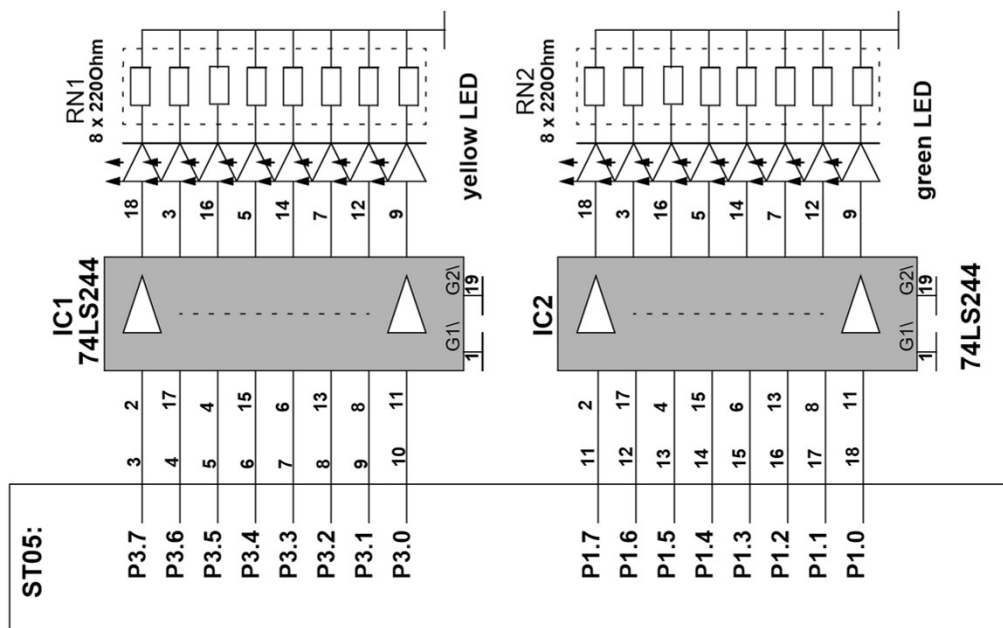
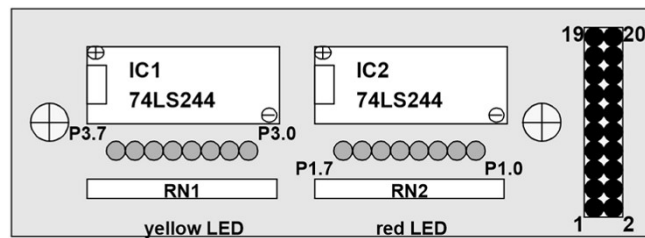
ST03/ST04-Pinning
(flatcable-bridge)

Function	Pin	Pin	Function
+5V	19	20	+5V
P1.1	17	18	P1.0
P1.3	15	16	P1.2
P1.5	13	14	P1.4
P1.7	11	12	P1.6
P3.1	9	10	P3.0
P3.3	7	8	P3.2
P3.5	5	6	P3.4
P3.7	3	4	P3.6
Gnd	1	2	Gnd

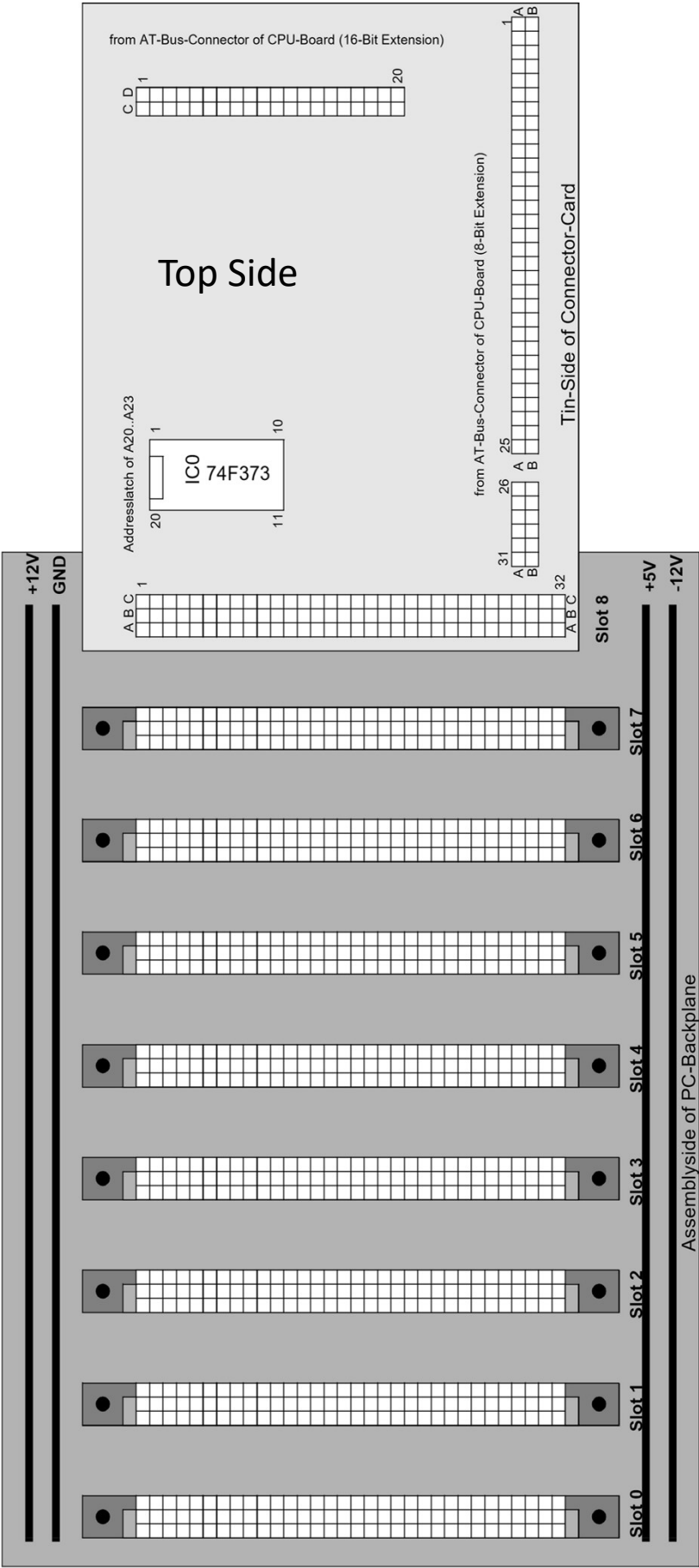
IO Front-Panel + LED Driver board + RS232-Connection



Assemblingside:



PC-AT Bus to 9x VG96-Backplane & Connector-Card



Mon51 Connectivity

**Tabelle 3: ST02: 25 Pin Sub-D Connector
for 8031/51-Port-Lines, (female)**

Functionality	Pin	Pin	
GND	25	13	GND
P1.1	24	12	P1.0
P1.3	23	11	P1.2
P1.5	22	10	P1.4
P1.7	21	9	P1.6
IN (Max232)	20	8	RXD (V24)
		7	TXD (V24)
OUT (Max232)	19	6	P3.0 (RxD TTL)
P3.1 (TxD)	18	5	P3.2 (Int0 low)
P3.3 (Int1 low)	17	4	P3.4 (T0)
P3.5 (T1)	16	3	P3.6 (WR\)
P3.7 (RD\)	15	2	Reset (Up)
+5V	14	1	+5V

A-Line	Stift	Stift	C-Line
GND	1	2	IRQ2
GND	3	4	IRQ3
GND	5	6	IRQ4
GND	7	8	IRQ7
GND	9	10	IRQ8
GND	11	12	GND
GND	13	14	OSC (14MHz)
GND	15	16	GND
GND	17	18	Refresh\
GND	19	20	TC
+5V	21	22	Master\
DACK7	23	24	DRQ7
DACK6	25	26	DRQ6
DACK5	27	28	DRQ5
DACK4	29	30	DRQ4
DACK3	31	32	DRQ3
DACK2	33	34	DRQ2
DACK1	35	36	DRQ1
DACK0	37	38	DRQ0
GND	39	40	GND

**Tabelle 1: ST01
PC-AT 64pin VG-Connector
(/w free B-line) (male)**

A-Line	Stift-Nr.	C-Line
GND	1	GND
D15	2	D7
D14	3	D6
D13	4	D5
D12	5	D4
D11	6	D3
D10	7	D2
D9	8	D1
D8	9	D0
MEMR\	10	IOCHRDY\
MEMW\	11	AEN
A23	12	A19
A22	13	A18
A21	14	A17
A20	15	A16
IORD\	16	A15
IOWR\	17	A14
SBHE	18	A13
MEMCS16\	19	A12
IOCS16\	20	A11
IRQ15	21	A10
IRQ12	22	A9
IRQ11	23	A8
IRQ10	24	A7
IRQ9	25	A6
IRQ5	26	A5
+12V	27	A4
BALE	28	A3
IOCHK\ = NMI\	29	A2
RESET\	30	A1
Clk (8MHz)	31	A0
+5V	32	+5V

**Tabelle 2: PC-AT-Expansion Connector,
upper (optional)**

Part List of MON51 board

Parttype	Number	Description	Functionality	Socket-Pinning
IC1	1	M8031/51	8-Bit M8031/51-CPU	DIP40
IC2	1	74HCT373	8031-Address-buffer, lower	DIP20
IC3-4, IC12-14	5	74HCT645, 74HCT245	ADdress-+DataLine-Driver	DIP20
IC5	1	27C64 ... 27C256	8031-ROM, <= 200ns	DIP28
IC6	1	xx64,43256	8031-Ram, <= 150ns	DIP28
IC7	1	MAX232	seriell V24-Line-driver	DIP16
IC8	1	74LS02		DIP14
IC9	1	74HCT244	Control-line-driver for Ram/Rom	DIP20
IC10	1	GAL20V8	GAL 15ns, for ADdressdecoding and Wait-Control	DIP24
IC11,15	2	74F138	Addressdecoding for Slot-Select/Port-Select	DIP16
IC15	1	74HCT04	6 x Inverter	DIP14
IC16	1	74ALS374	Control-Port for Reset/Int0/Int1	DIP20
Clock51	1	Quarz-Osc.	11.0592 MHz 8031/51-Clock	DIP14
T1	1	BC238	Inverter of IOChRdy-Line, open Collector	TOx
C1	1	10uF\10V	Elko for Reset-Timing	standing
C2	1	220uF\16V	Buffer-Elko for 5V-Line	standing
C3,4	2	47uF/25V	for MAX232	standing
C5,6	2	22uF/25V	for MAX232	standing
R1	1	3.3k\1/8W	Pull-Up for Slot-Select-Jumper-Line	Cole. 10%
R2	1	470\1/8W	Driver-resistor for Reset-Timing	Cole. 10%
R3	1	10kOhm	Wait/-Inverter-functionality	Cole. 10%
R4,5	2	1kOhm	Pullup for Int0/Int1	Cole. 10%
J1,2	2	3 Pin-Line	Jumper for Ram/Rom-Selection	3Pin, gold
J3	1	2x8Pin-Line	8 Jumper for Card-Base-Address-Selection	16Pin, gold
ST01	1	64PinVG	PC386-AT-Connector 64Pin, male a+c , without line B!	90 degree
ST02	1	25Pin-SubD	8031-Port-Line-connector, 25 Pin SubD-female, print	90degree
ST03-04	2	2x10PinLine	Bridge-connection for Port-Line from CPU to ST02	20Pin, gold
mainboard	1	Euro-card	Print-Board, 15.8 x 10 cm, EuroCard	
Front	1	Front-plate	Front-Panel-Plate of ALU for 19"-assembling	ALU
SW1	1	Touchswitch	Reset-Switch for 8031/51-CPU	scrueLLing
IC0	1	74F373	fast 4-Bit Addresslatch for AT-Bus-S-Addresslines	DIP20

GAL-File: M51.GAL for 8051/AT-Handshake Control

Diese Datei enthält den Sourcecode fuer einen Waitstate-Generator am AT-Bus-System.
 Der Generator, basierend auf einem GAL20V8, wartet auf ein inaktives Slave(8051)-Bussystem:
 -> Bei M8051 ist das Signal ALE = '1', und blockiert damit den Prozessortakt der Slave-CPU
 solange, bis der AT-Bus-Zyklus beendet ist.
 -> Das WAIT-Signal zum AT-Bus wird solange erzeugt (IOCHRDY\) wie die Slave-Karte selektiert
 ist und der AT-Bus-Zyklus noch aktiv ist. Dabei wird per internem Zähler =3 (bei 8MHz) eine
 Zugriffszeit von 250 ns generiert.
 Die maximale Auflösung im Adressraum ist 1MByte, so dass für jeden Slot (8x) 1Mbyte zur
 Verfügung steht.
 Das GAL teilt dieses nochmal in 2 x 512kByte Select-Signale für SDRam51 und EPROM51 auf.
 Die Basisadresse eines jeden Slots lautet:

$$0x800000 + n * 0x100000 \quad (\text{für } n = \text{Slot-ID } 0..7)$$

Letzte Aenderung: 2.7.1996 V1.0

Wie üblich beginnt der GALProg Sourcecode mit dem Kommando:

CHIP	WAIT_STATE_ATSLAVE	GAL20V8
CLK	BALE	MEMRD
/OE	IN11	/SLOTSEL
HIT =	/HITIN * /BALE * /MEMRD + /HITIN * /BALE * /MEMWR	; = 1 wenn Slota-Base-address erkannt ; worden ist (externer 74F138): ; (BA20, BA21, BA22) = SlotId 0..7 ; BA23=1, BAEN=0, 1MB-range per slot
ROMSEL =	HIT * /BA19 * /MEMRD * /BA18	; /CE fuer Slavecard-ROM, 256kB-range
RAMSEL =	HIT * /BA19 * BA18	; /CE fuer Slavecard-RAM, 256kB-range
SLOTSEL:=	HIT * ALE51	; =1 wenn Slave-Zyklus beendet ist ; und AT-Bus-Zyklus beginnen kann ; dient als Switch der Bustreiber
CLKOUT =	/CLKIN * /SLOTSEL	; Slave-CPU-Clock Anschluss: 12 MHz
BIT0 :=	HIT * /BIT0 + HIT * BIT0 * BIT1	; Zaehlstufe fuer Waitzyklus, lower
BIT1 :=	HIT * BIT0 * /BIT1 + HIT * /BIT0 * BIT1 + HIT * BIT0 * BIT1	; Zaehlstufe fuer Waitzyklus, higher ; auf 3 stehen bleiben bis Zyklusende
WAIT =	HIT * /BIT0 * /BIT1 + HIT * BIT0 * /BIT1 + HIT * /BIT0 * BIT1	; Wait-Signal: ; HIT -> solange slavecard-select =1 ; und Zaehler in Stufe 0..2

GAL Prog update 30.7.1996

MON51.GAL:

=====

Diese Datei enthaelt den Sourcecode fuer einen Waitstate-Generator eines AT-Bus-Systems. Der Generator, basierend auf einem GAL20V8, wartet auf ein inaktives Slave-Bussystem:

-> bei M8051 ist ALE = '1', und blockiert dann den Prozessortakt der Slave-CPU solange, bis der AT-Bus-Zyklus beendet ist.

Das WAIT-Signal zum AT-Bus wird solange erzeugt (IOCHRDY\), wie die Slave-Karte selektiert ist und der AT-Bus-Zyklus noch aktiv ist. Dabei wird per internem Zaehler (auf 3 bei 8MHz) eine Zugriffszeit von 250 ns generiert.

Die maximale Aufloesung im Adressraum ist 1MByte, so dass fuer jeden Slot (8x) je 1MByte zur Verfuegung steht.

Das GAL teilt dieses Adressfenster nochmal in 2 x 512kByte Selekt-Signale fuer SDRam51 und EProm51 auf.

Die Basisadresse eines jeden Slots lautet somit:

$0x800000 + n * 0x100000$ (fuer n = Slot-ID 0..7)

Letzte Aenderung: 28.7.1996 V1.1

Wie ueblich beginnt der Sourcecode mit dem Kommando

```
CHIP    WAIT_STATE_ATSLAVE    GAL20V8
;=====
CLK    BALE    MEMRD MEMWR    HITIN    ALE51    BA19    CLKIN    IN8    IN9    IN10    GND

/OE    IN11    /LOCAL WAIT    /ROMSEL /RAMSEL    HIT    CLKOUT    BIT0    BIT1    IN12    VCC
;=====

HIT      =      /HITIN * BALE * MEMRD      ; = 1 wenn Slota-Base-address erkannt
            + /HITIN * BALE * MEMWR      ; worden ist (externer 74F138):
            ; BA20, BA21, BA22, BA23=1, BAEN=0

ROMSEL   =      HIT * /BA19 * MEMRD      ; /CE fuer Slavecard-ROM
RAMSEL   =      HIT * BA19                ; /CE fuer Slavecard-RAM
LOCAL    =      HIT * ALE51                ; =1 wenn Slave-Zyklus beendet ist
            ; und AT-Bus-Zyklus beginnen kann
            ; dient als Switch der Bustreiber

CLKOUT   =      CLKIN * /LOCAL            ; Slave-CPU-Clock Anschluss: 12 MHz
BIT0     :=      /BIT0 * LOCAL            ; Zaehlstufe fuer Waitzyklus, lower
BIT1     :=      BIT1 * /BIT0 * LOCAL     ; Zaehlstufe fuer Waitzyklus, higher
            + /BIT1 * BIT0 * LOCAL

WAIT     =      HIT * /BIT0 * /BIT1      ; Wait-Signal:
            + HIT * BIT0 * /BIT1        ; HIT -> solange slavecard-select =1
            + HIT * /BIT0 * BIT1        ; und Zaehler in Stufe 0..2
;WAIT.TRST = HIT * /BIT0 * /BIT1        ; das ganze fuer Tristate-Ausgang
;
;      + HIT * BIT0 * /BIT1            ; nochmal
;      + HIT * /BIT0 * BIT1
```