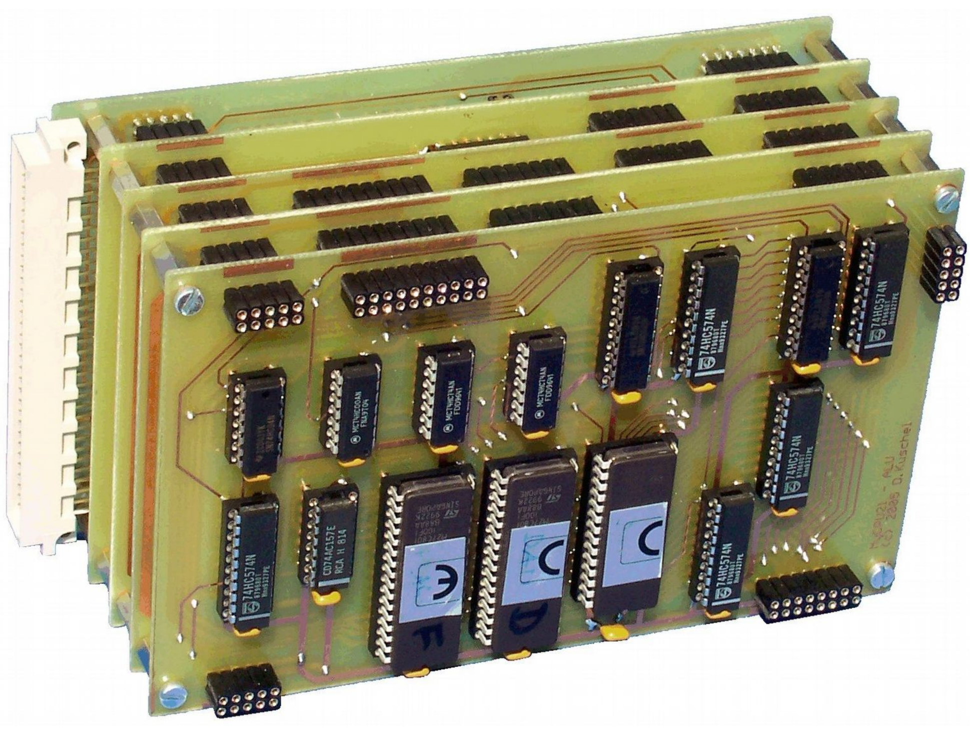
MyCPU 2.3

- Selfbuild Guide -



Dennis Kuschel · Corintostraße 21 · 28279 Bremen · Germany · mycpu.eu · dennis_k@freenet.de 2015-07-23

MyCPU23 Technical Details

- 8-bit processor, 8-bit internal and external data bus
- 16 address lines, 64kb addressable RAM and 64kb addressable ROM memory, 128kb addressable memory total
- Architecture similar to Harvard (separate code and data memory, but only one common address and data bus)
- Maximum processor core speed is around 10 MHz when 74ACxxx gates and fast EPROM's are used, but 8 MHz is guaranteed, and at least 4 MHz can be reached with the cheaper 74HCxxx gates. Save operation is still possible with 74HCxxx gates and a 20MHz oscillator (core frequency approx. 6,7 MHz).
- 1 maskable hardware interrupt (IRQ)
- 5 x 8-bit general purpose register
- lookup-table based ALU for maximum speed
- 61 integrated circuits (55 logic gate IC's and 6 EPROM's)
- User definable microcode, the microcode for a 6502 like CPU is provided
- Voltage operating range is 4.75V – 5.25V
- Power consumption is max. 150mA at 5V = 0.75W
- **up to 20% faster than MyCPU22**

MyCPU23 Microcode Details

- The instruction set is similar to that one of the 6502, but is not binary compatible
- 256 OP-Codes
- 14 Addressing Modes: immediate 16-bit, immediate 8-bit, immediate zeropage, direct absolute, direct zeropage, direct absolute plus index, indirect absolute, indirect zeropage, indirect absolute plus index, indirect zeropage plus index, indirect plus index absolute, absolute pointer 16-bit, absolute pointer 8-bit, immediate registers
- 1 Software Interrupt
- 256 byte call stack, 8-bit stack pointer
- 3 8-bit general purpose registers: accu, x- and y- index-register. The x- and y- register can be put together to a 16-bit data pointer register.
- User performance index: 6.9 core clock cycles per OP-Code in average, resulting in a speed of 1.16 MIPS at 8 MHz processor core frequency.

Contents

Figures

Tables

Author:

Dennis Kuschel
Corintostraße 21
28279 Bremen, Germany

web : http://www.mycpu.eu
email: dennis_k@freenet.de

IMPORTANT INFORMATION
about stability issues

The design uses several 74**AC**xxx - gates. For a proper operation it is recommended to replace all 74**AC**xxx - parts through 74**HC**xxx - parts. This modification reduces the maximum operation frequency, but it increases the system stability. With this configuration you should be able to reach at least 5.7 MHz. Please see chapter 1.5.4.4 for details. Note: If you encounter stability problems, it is always a good idea to try other settings for JP4.

WARNING
For a successful reproduction of a MyCPU it is very important to use the right type of 10nF and 100nF capacitors:

USE ceramic capacitors only:

DO NOT USE foil or film capacitors:

UPGRADE INFORMATION

If you have a MyCPU v2.1 or v2.2 you can upgrade to MyCPU v2.3. Please upgrade the boards "Interface and Registers" and "Microcode Control". The other boards do not have changed, thus the ALU, the Program Counter and the Decoder Board can be re-used. But on the Decoder and on the Program Counter Board the population has changed, several 74ACxxx parts where replaced by 74HCxxx types.
Important: The content of the microcode EPROM's on the Microcode Control Board has changed. You need to re-program the EPROM's IC1, IC2 and IC3.

1 Boards

1.1 Arithmetic Logic Unit (ALU)

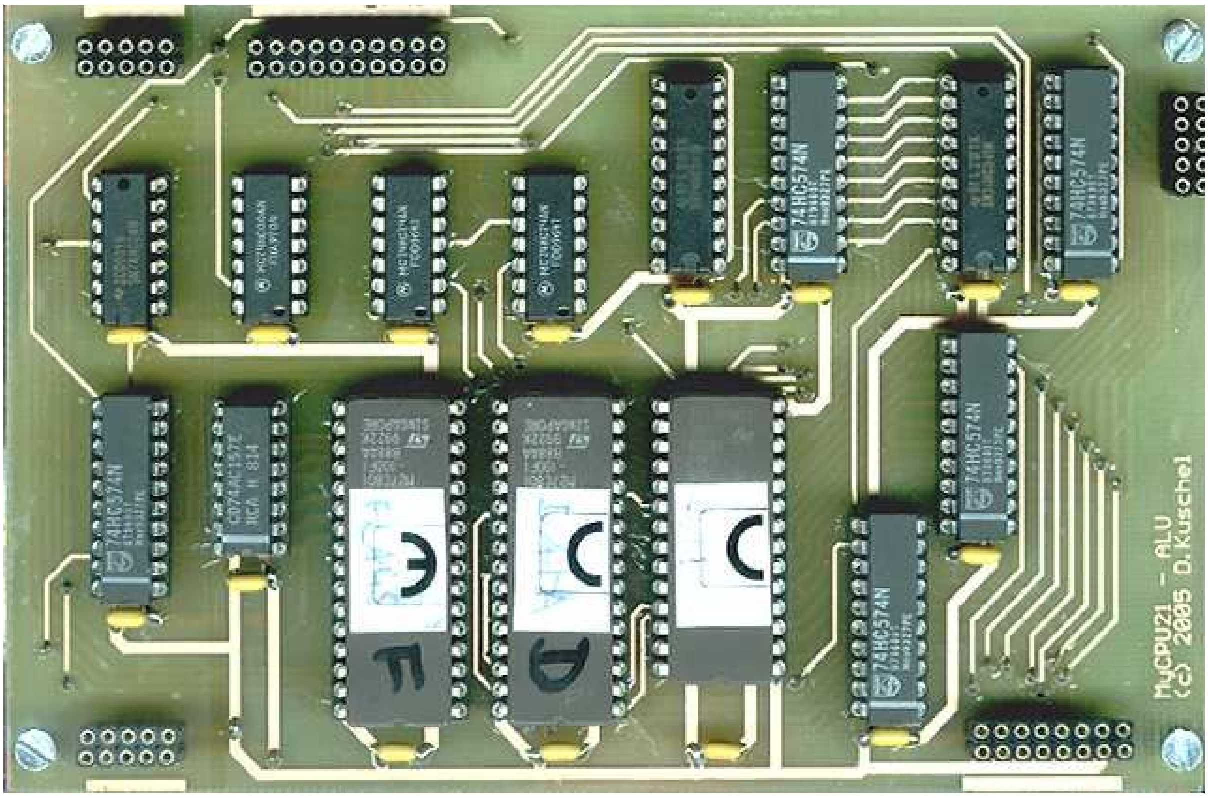


Fig. 1: Arithmetic Logic Unit (ALU) Board

1.1.1 Description

The Arithmetic Logic Unit is the part of the Microprocessor that does all the mathematics calculations. The ALU is based on lookup-tables to keep the design as simple as possible. The unit supports 16 arithmetic operations, where eight operations take two 8-bit-operands and eight operations take only one operand. The result of every operation is also 8-bit wide. The ALU supports three flags: Carry, Sign and Overflow. Since the ALU is table-based, the slowest operation takes only three clock cycles: Two clock cycles are needed to load both operands, and one more cycle is used to store the result in the destination register.

1.1.2 Placement of Components

After you have soldered all Via's, you can continue with the integrated circuits. I suggest you not to use sockets for the IC's, except for the EPROM's. If you wish to use sockets for all IC's, you must use precision sockets. Only the high quality sockets allow you to solder the pads on the top side of the board. I strongly recommend you to follow the placement order I have noted in the placeplan below (see blue numbers, and start with the IC that has the blue number 1). When all IC's are placed and soldered, you can continue to place the capacitors. In the last round the board connectors are placed and soldered.

ATTENTION!
Please be careful, and don't forget to solder a pad on the top side of the board. I have marked all critical pads with red colour in the placeplan below. Please check if you have really soldered these pads!

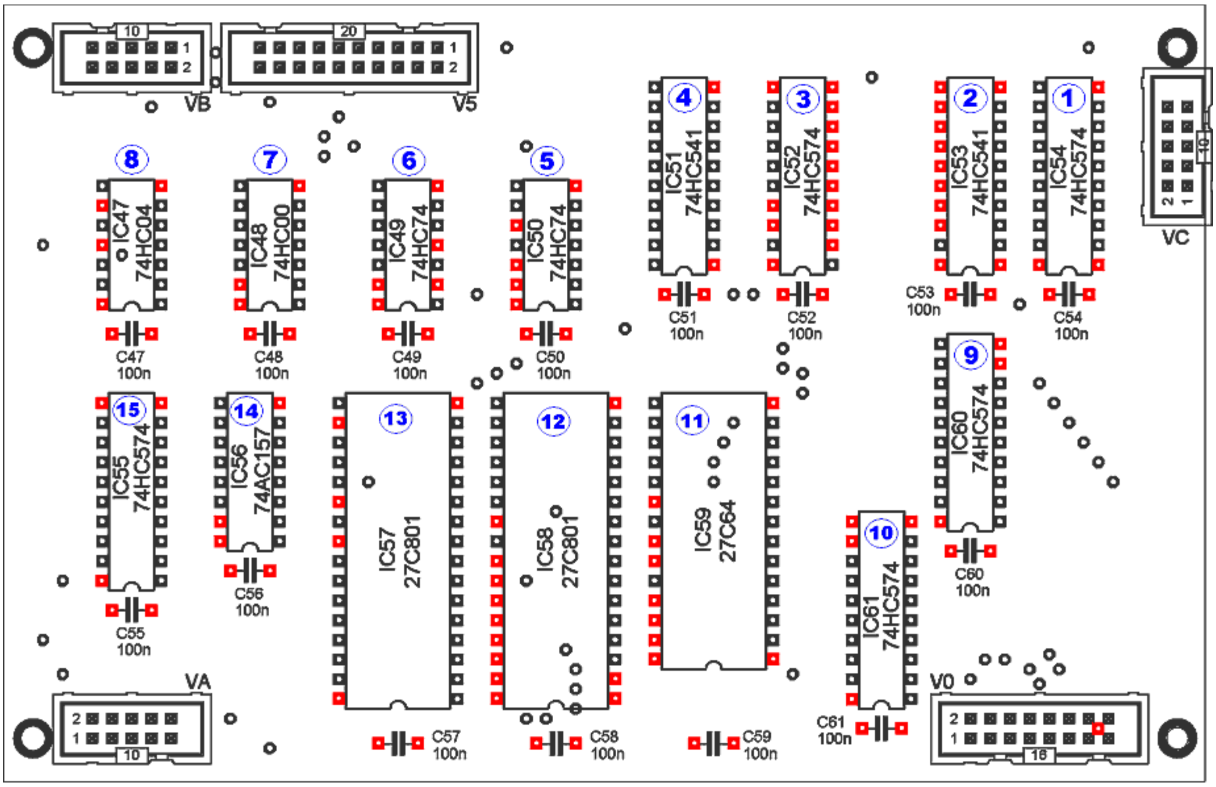


Fig. 2: Arithmetic Logic Unit - Placeplan

1.1.3 Partlist

74HC00	IC48
74HC04	IC47
74HC74	IC49, IC50
74AC157	IC56
74HC541	IC51, IC53
74HC574	IC52, IC54, IC55, IC60, IC61
27C64 **-or-** 27C256, 100ns	IC59
27C801 **-or-** 27C080, 100ns	IC57, IC58
100nF ceramic capacitor	C47, C48, C49, C50, C51, C52, C53, C54, C55, C56, C57, C58, C59, C60, C61
10 pin header	VA, VB, VC
16 pin header	V0
20 pin header	V5

Note: If you are using the fast 100ns EPROM's and you will not clock the CPU core higher than 8 MHz, the 74AC157 can be replaced by the slower 74HC157.

1.2 Microcode Control

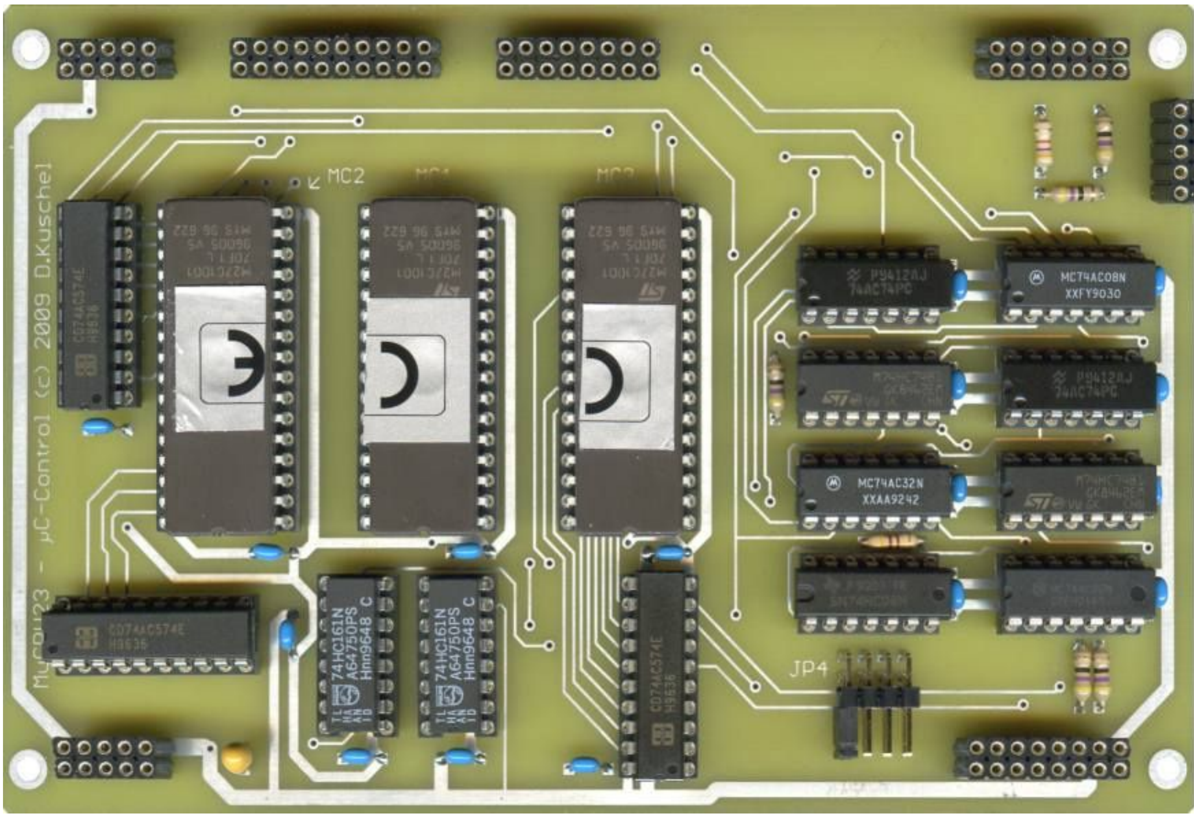


Fig. 3: Microcode Control Board

1.2.1 Description

The control board is the core of the CPU. It contains the big state-machine the controls all components of the CPU. The state machine is implemented as a lookup-table with feedback lines. The lookup-table is built with three 128kb*8 EPROM's. The 8 bits of the OP-code, five counter lines, the three flag lines and the interrupt line are the inputs for the lookup table. The 24 output lines of the EPROM's steer the several components of the CPU. Commonly said, the three EPROM's contain the microcode of the CPU. The microcode can be seen as an interpreter for the OP-code. A microcode can have a depth of 32 micro instructions, but in common only 10 micro instructions are required to interpret an OP-code.

1.2.2 Placement of Components

After you have soldered all Via's, you can continue with the integrated circuits. I suggest you not to use sockets for the IC's, except for the EPROM's. If you wish to use sockets for all IC's, you must use precision sockets. Only the high quality sockets allow you to solder the pads on the top side of the board. I strongly recommend you to follow the placement order I have noted in the placeplan below (see blue numbers, and start with the IC that has the blue number 1). When all IC's are placed and soldered, you can continue to place the capacitors and resistors. In the last round the jumper block and the board connectors are placed and soldered.

ATTENTION!
Please be careful, and don't forget to solder a pad on the top side of the board. I have marked all critical pads with red colour in the placeplan below. Please check if you have really soldered these pads! DO NOT USE SOCKETS FOR IC9 - IC16 !

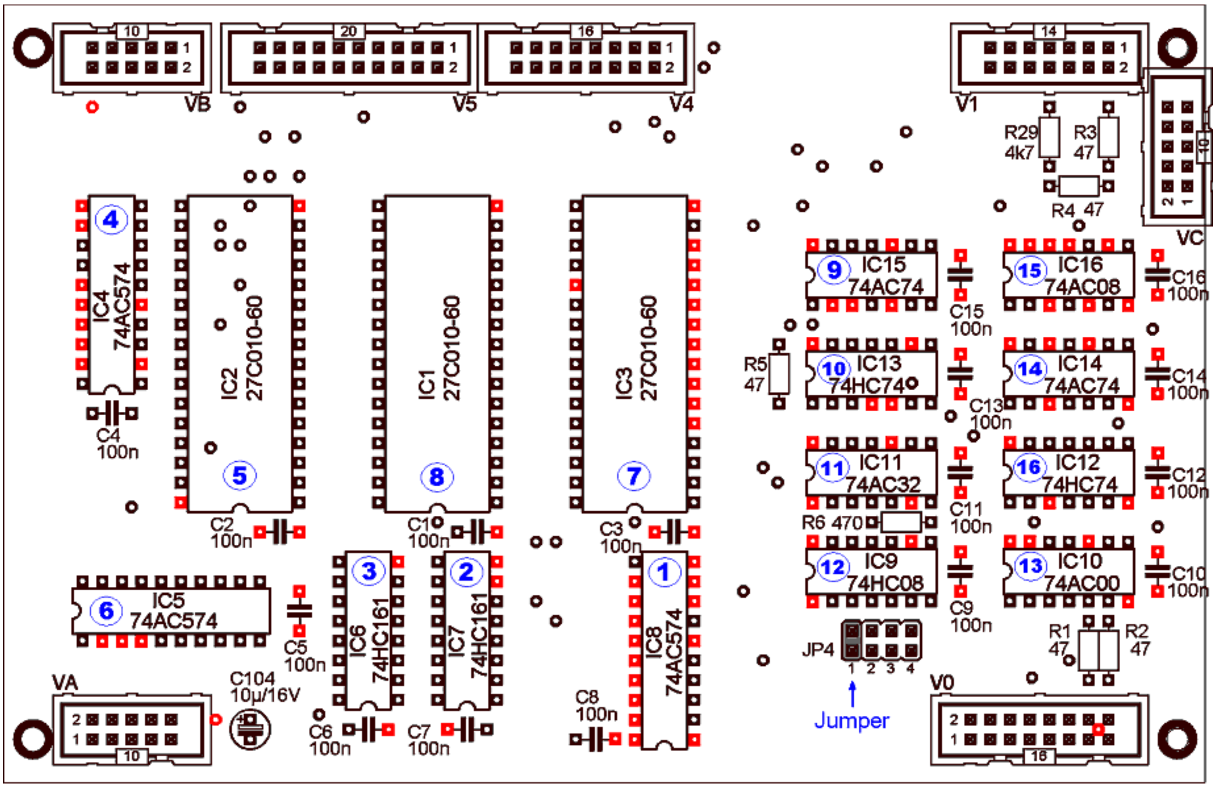


Fig. 4: Microcode Control – Placeplan

1.2.3 Partlist

74AC00	IC10	
74AC08	IC16	
74HC08	IC9	*Must be a 74**HC**08*
74AC32	IC11	
74AC74	IC14, IC15	
74HC74	IC12, IC13	
74HC161	IC6, IC7	
74AC574	IC4, IC5, IC8	
27C010 **-or-** 27C1001, 60ns	IC1, IC2, IC3	
100nF ceramic capacitor	C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12, C13, C14, C15, C16	
10µF / 16V, tantalum	C104	
47 Ohm	R1, R2, R3, R4, R5	
470 Ohm	R6	
4.7 kOhm	R29	
10 pin header	VA, VB, VC	
14 pin header	V1	
16 pin header	V0, V4	
20 pin header	V5	
2*4 pin jumper block + jumper	JP4	

1.2.4 Jumper Settings

The Jumper JP4 is used to change the runtime compensation of the EPROM delay. This is required if the CPU shall run on high clock frequencies (6 MHz and above). Usually the jumper can be set to position 1 (like shown in the figures). But if you observe instabilities or crashes, you may try other jumper positions (2 trough 4) to fix a possible problem with the EPROM delay time.

Please keep the jumper JP4 in mind when you have taken MyCPU extension boards into service and you observe instabilities. The more components are connected to MyCPU, the less stable the MyCPU may work. You can try to compensate this with JP4.

1.3 Signal Decoder

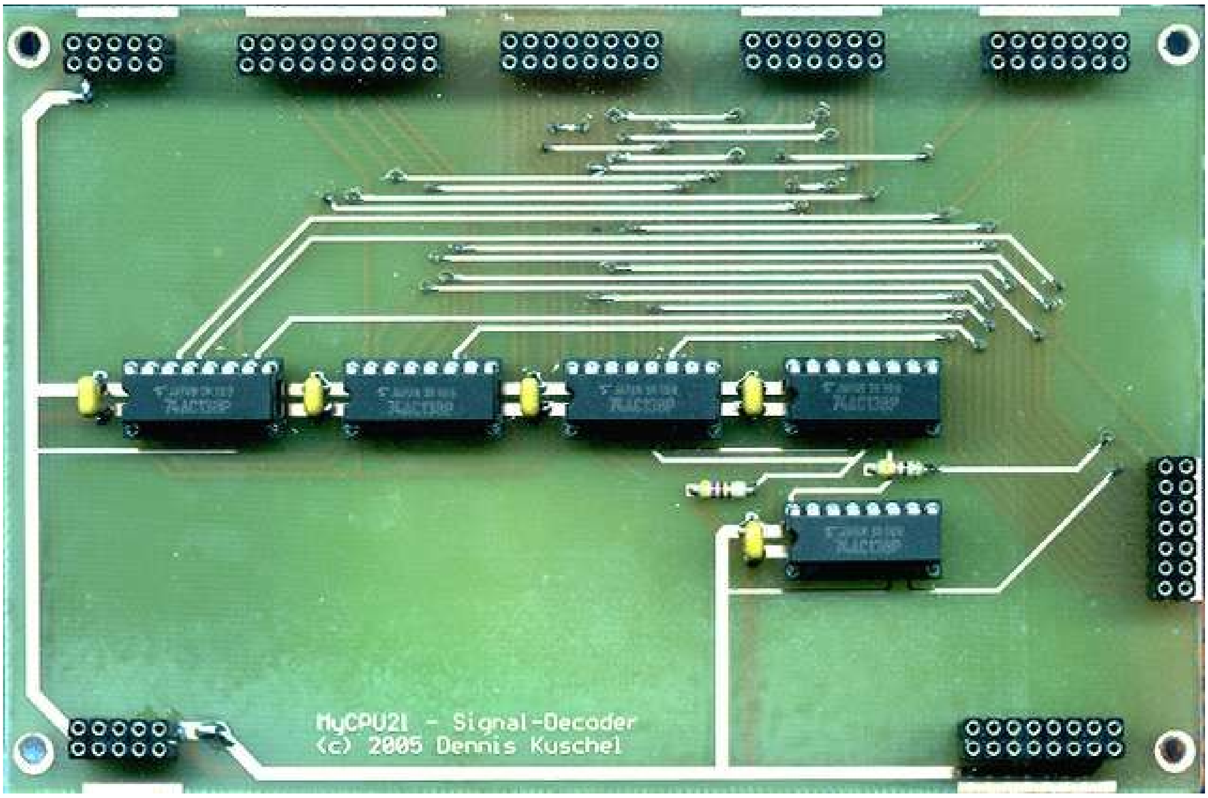


Fig. 5: Signal Decoder Board (Prototype, R30+R31 missing)

1.3.1 Description

The signal decoder board belongs to the CPU microcode control board. The eight output lines of EPROM IC1 are demultiplexed on this board. This board generates 16 read- and 16 write-control-lines. These lines are used to read data from and write data to registers.

1.3.2 Placement of Components

After you have soldered all Via's, you can continue with the integrated circuits. I strongly recommend you to follow the placement order I have noted in the placeplan below (see blue numbers, and start with the IC that has the blue number 1). When all IC's are placed and soldered, you can continue to place the capacitors and resistors. In the last round the board connectors are placed and soldered.

ATTENTION!
Please be careful, and don't forget to solder a pad on the top side of the board. I have marked all critical pads with red colour in the placeplan below. Please check if you have really soldered these pads!

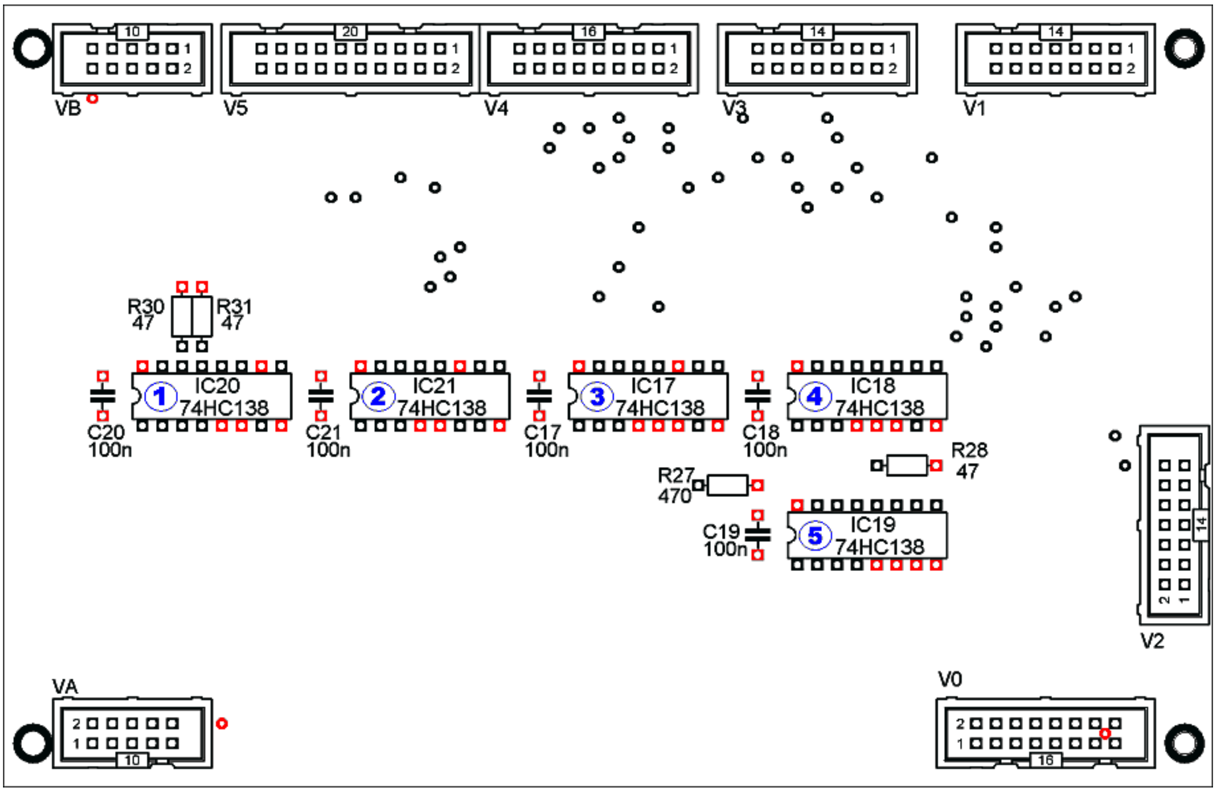


Fig. 6: Signal Decoder – Placeplan

1.3.3 Partlist

74HC138	IC17, IC18, IC19, IC20, IC21
47 Ohm	R28, R30, R31
470 Ohm	R27
100nF ceramic capacitor	C17, C18, C19, C20, C21
10 pin header	VA, VB
14 pin header	V1, V2, V3
16 pin header	V4
20 pin header	V5

1.4 Program Counter

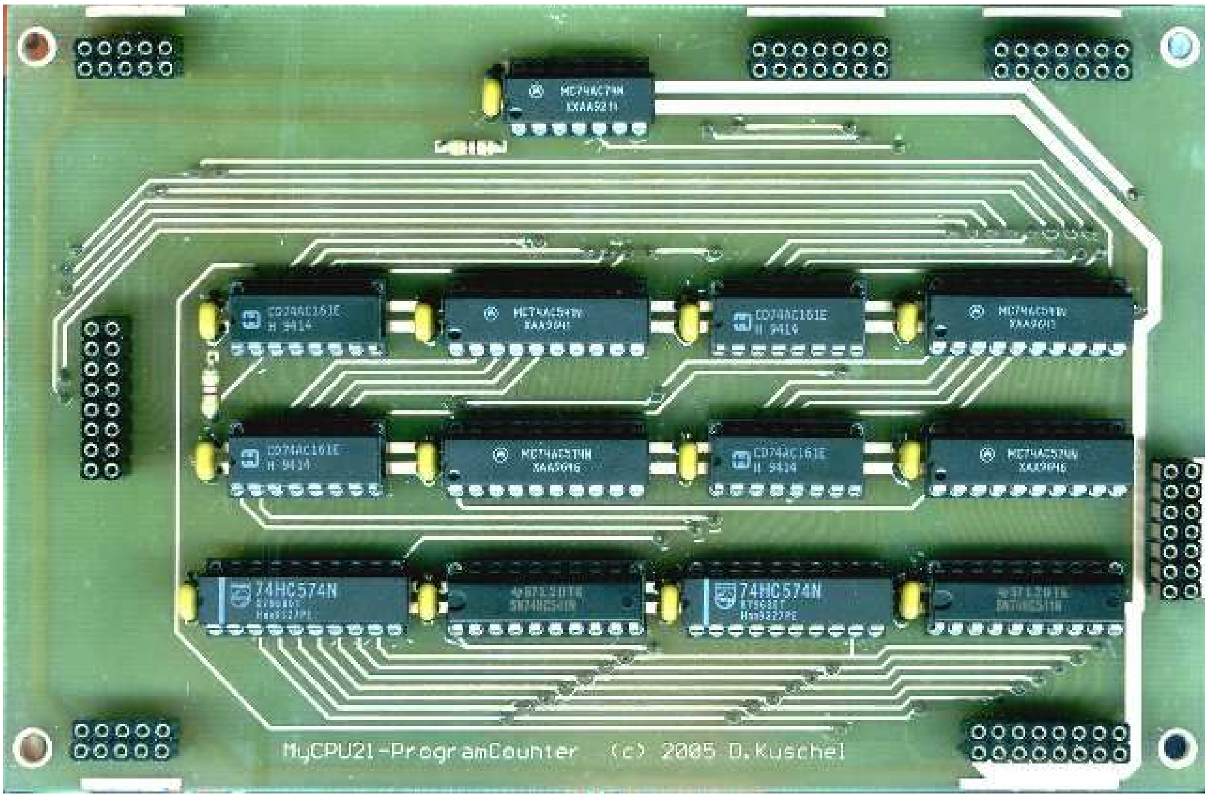


Fig. 7: Program Counter Board

1.4.1 Description

The board contains the program counter (PC) and the rest of the address line logic. The program counter is a 16-bit wide counter that is built with four 4-bit counters (74HC161). The counters can be loaded with a new initial address and the actual counter state can also be read back. Further more there is a logic that allows the CPU to access random addresses.

1.4.2 Placement of Components

After you have soldered all Via's, you can continue with the integrated circuits. I strongly recommend you to follow the placement order I have noted in the placeplan below (see blue numbers, and start with the IC that has the blue number 1). When all IC's are placed and soldered, you can continue to place the capacitors and resistors. In the last round the board connectors are placed and soldered.

ATTENTION!
Please be careful, and don't forget to solder a pad on the top side of the board. I have marked all critical pads with red colour in the placeplan below. Please check if you have really soldered these pads!

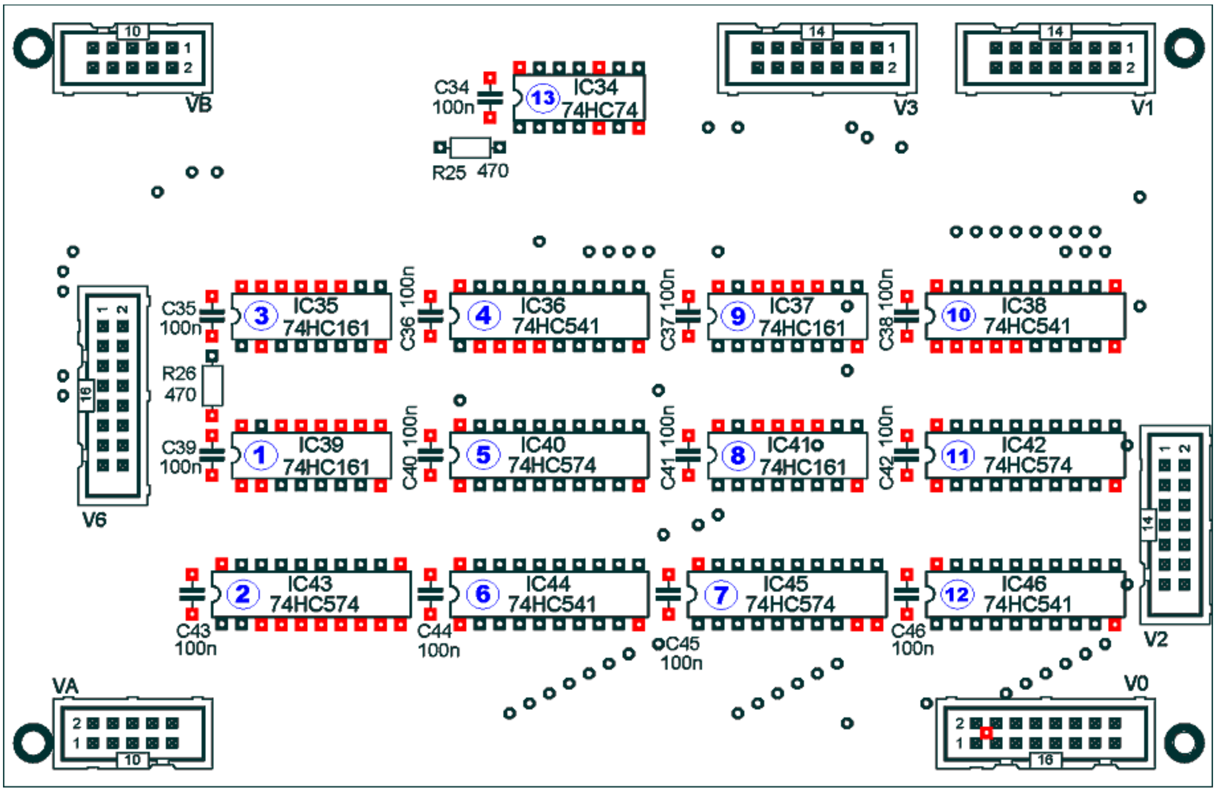


Fig. 8: Program Counter – Placeplan

1.4.3 Partlist

74HC74	IC34
74HC161	IC35, IC37, IC39, IC41
74HC541	IC36, IC38, IC44, IC46
74HC574	IC40, IC42, IC43, IC45
100nF ceramic capacitor	C34, C35, C36, C37, C38, C39, C40, C41, C42, C43, C44, C45, C46
470 Ohm	R25, R26
10 pin header	VA, VB
14 pin header	V1, V2, V3
16 pin header	V0, V6

1.5 Interface and Registers

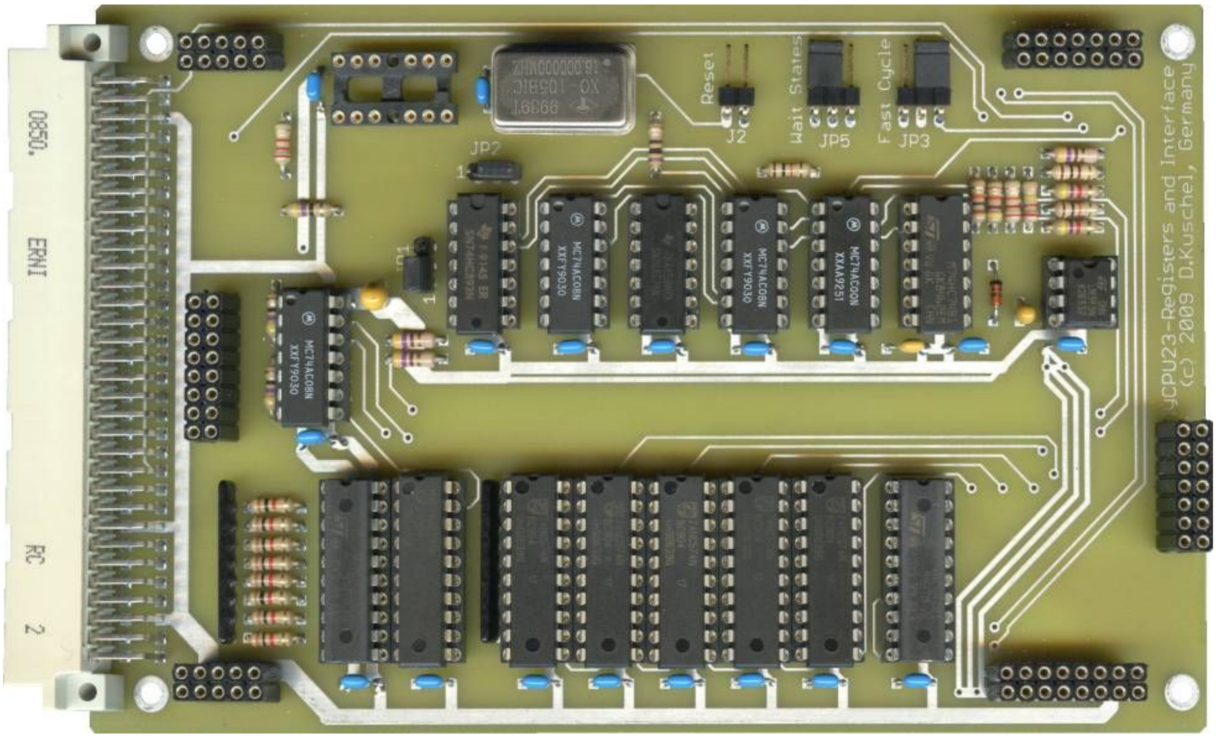


Fig. 9: Interface and Register Board

1.5.1 Description

The board contains the core-registers ACCU / X-Reg / Y-Reg, the stack pointer register and a temporary register, the bus interface driver, a clock generating- and control-circuit, and the reset logic.
The board has two sockets for two different oscillators, whereas only one oscillator is required and the second one is optional. For possible oscillator frequencies and jumper settings please see chapter 1.5.4.

1.5.2 Placement of Components

After you have soldered all Via's, you can continue with the integrated circuits. I strongly recommend you to follow the placement order I have noted in the placeplan below (see blue numbers, and start with the IC that has the blue number 1). When all IC's are placed and soldered, you can continue to place the capacitors and resistors. In the last round the board connectors and jumpers are placed and soldered.

ATTENTION!
Please be careful, and don't forget to solder a pad on the top side of the board. I have marked all critical pads with red colour in the placeplan below. Please check if you have really soldered these pads!

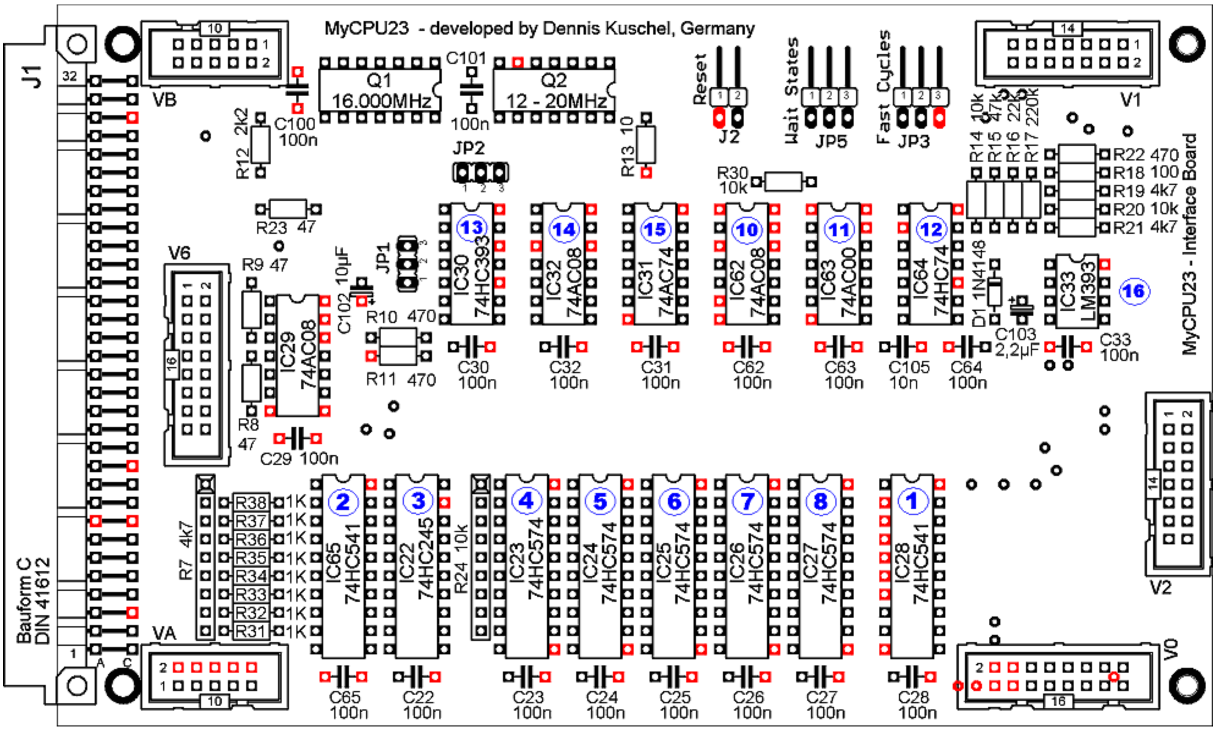


Fig. 10: Interface and Register – Placeplan

1.5.3 Partlist

74AC00	IC63
74AC08	IC29, IC32, IC62
74AC74	IC31
74HC74	IC64
74HC245	IC22
74HC393	IC30
74HC541	IC28, IC65
74HC574	IC23, IC24, IC25, IC26, IC27
LM393	IC33
10nF ceramic capacitor	C105
100nF ceramic capacitor	C22 - C33, C62 - C65, C100, C101
2.2µF / 16V, tantalum	C103
10µF / 16V, tantalum	C102
10 Ohm	R13
47 Ohm	R8, R9, R23
100 Ohm	R18
470 Ohm	R10, R11, R22
1 kOhm	R31, R32, R33, R34, R35, R36, R37, R38
2.2 kOhm	R12
4.7 kOhm	R19, R21
10 kOhm	R20, R30, R14
22 kOhm	R16
47 kOhm	R15

220 kOhm	R17
SIL 8 x 4,7 kOhm	R7
SIL 8 x 10 kOhm	R24
D1	1N4148
Crystal Oscillator	Q1, Q2
3 pin jumper	JP1, JP2, JP3, JP5
2 pin connector (with cable)	J2
DIN 41612 Connector	J1
10 pin header	VA, VB
14 pin header	V1, V2
16 pin header	V0, V6

1.5.4 CPU Frequency Jumper Settings

The figure below describes the various jumpers on the interface board:

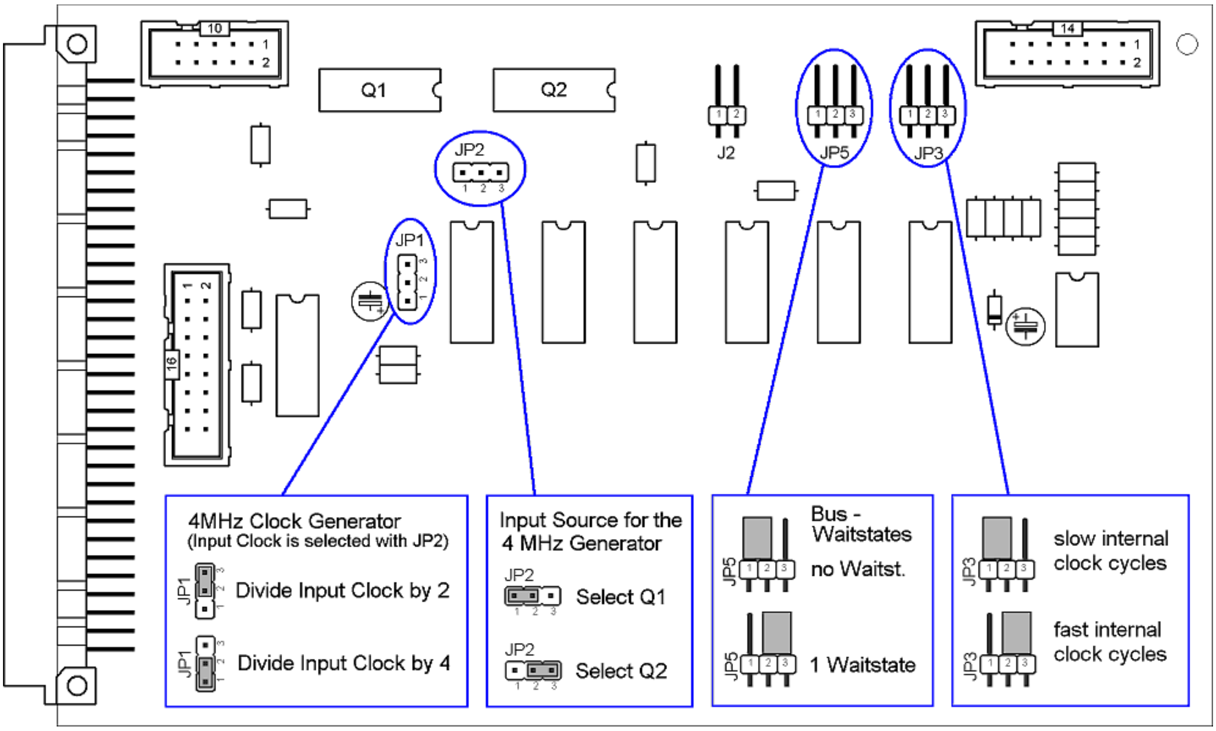


Fig. 11: Clock Control - Jumper Settings

Default jumper settings: JP1: [1-2], JP2: [2-3], JP3: [2-3], JP5: [2-3]
Default oscillator population: Q1: not populated, Q2: populated with 16MHz oscillator

1.5.4.1 Oscillator population

Q1 and Q2 shall be placed with a socket. You should use a precision DIL socket (14 pins) for the oscillators. Before soldering the sockets, the pins 4 and 11 should be removed (cut with a knife). This enables you to fix the oscillator in the socket with a cable strip.

Q1 must be populated when Q2 has an other frequency than 8.0000 MHz or 16.0000 MHz. Then Q1 must be populated with either 8.0000 MHz (set JP1 to 2-3) or 16.0000 MHz (set JP1 to 1-2). Don’t forget to set JP2 to position 1-2.

If Q1 is not populated, Q2 must have a frequency of either 8.0000 MHz or 16.0000 MHz, and JP2 must be set to position 2-3, and JP1 to 2-3 for an 8 MHz oscillator or to 1-2 for a 16 MHz oscillator.

If Q1 is populated, Q2 can have any frequency.

Summary:

- JP1 sets the clock divider for the 4.000 MHz bus clock
- JP2 chooses the clock source for the 4.000 MHz bus clock (either Q1 or Q2)

For the first run please use always the default configuration (see red box on previous page). When you got your MyCPU running with this configuration, you can start tweaking your MyCPU according to chapter 1.5.4.4.

1.5.4.2 Bus Wait States

Jumper JP5 can be used to disable bus wait states. Note that bus wait states are enabled by default. You may disable bus wait states to tune the MyCPU, but this requires really fast periphery like fast memories and a short backplane bus. With VGA-Unit or IDE-Controller attached you should always leave bus wait states enabled.

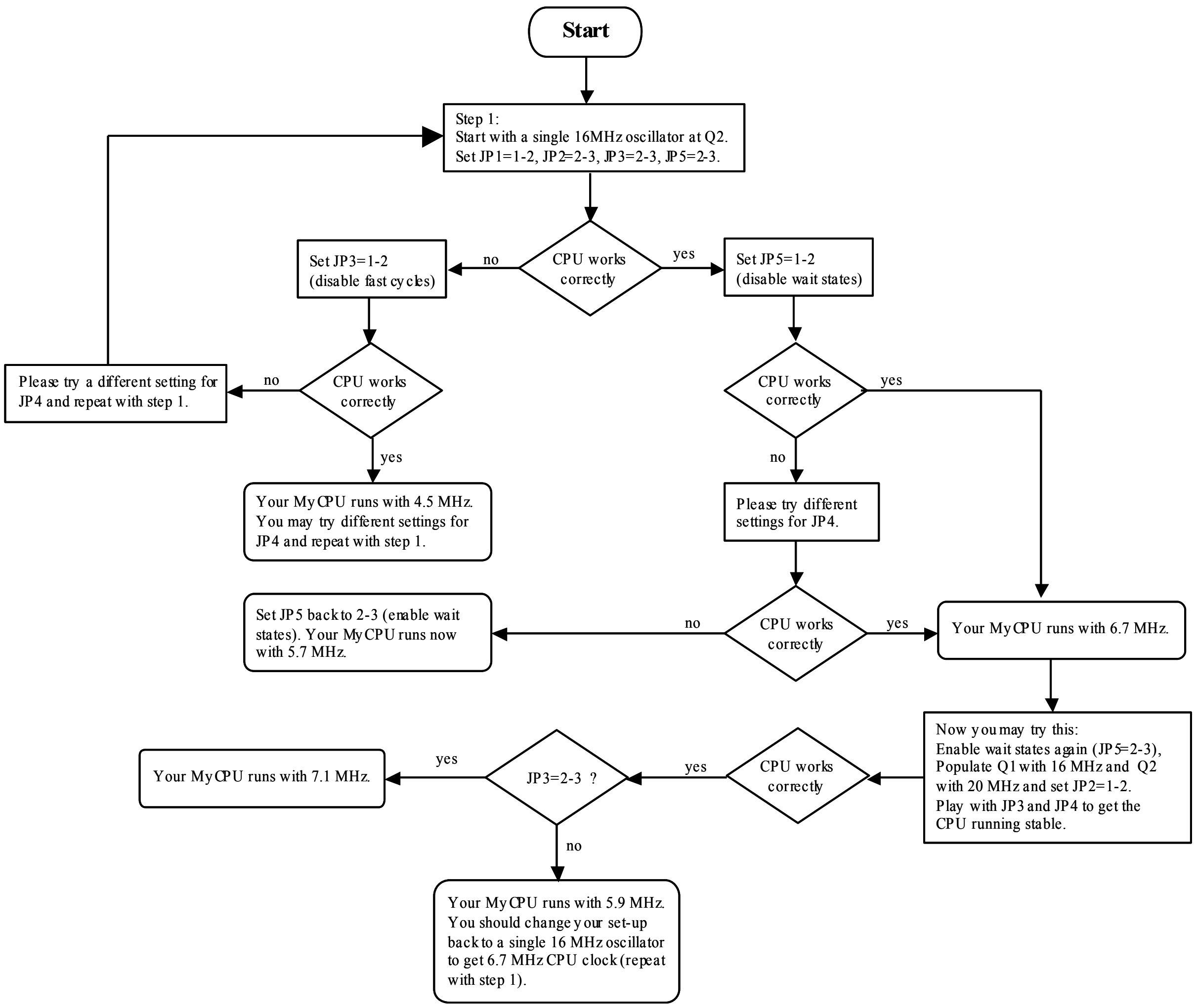
1.5.4.3 Fast internal clock cycles

By default MyCPU uses faster (that means shorter) clock cycles for internal register-to-register transfers. If you are using very slow EPROM's for the Microcode or you could not procure 74ACxxx-chips, it may be necessary to disable fast clock cycles. To slow down the internal clock, please set JP3 to position 1-2.

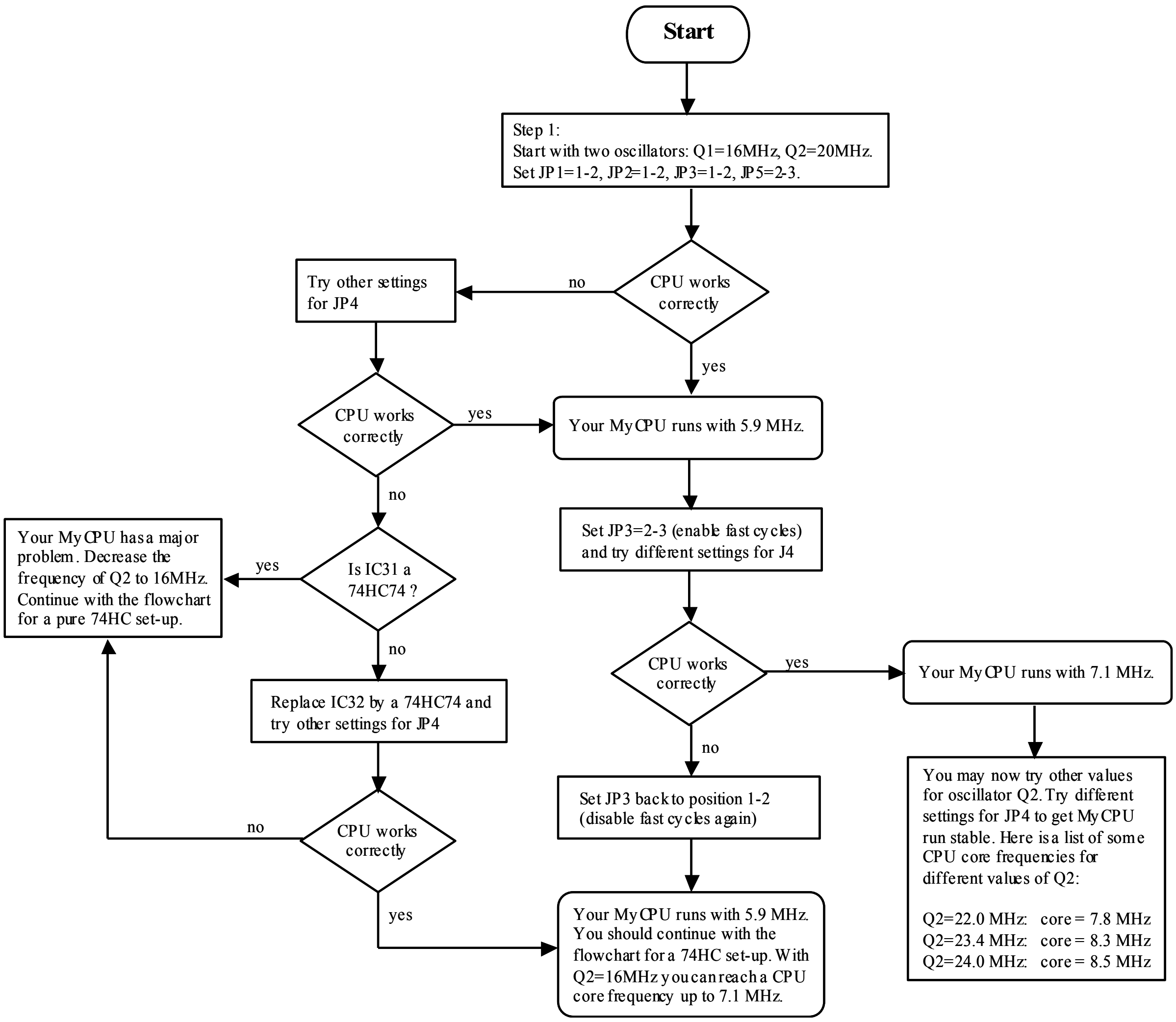
1.5.4.4 Tuning the MyCPU clock frequency

You can try to increase the CPU core clock frequency by using other jumper settings and by increasing the oscillator frequency. The following flowcharts will help you. You should use the first flow chart if you could not obtain all the required 74ACxxx parts. Use the second flowchart when you have populated the boards with 74ACxxx chips where they are required.

Use this flowchart to tweak you MyCPU if you have a pure 74HCxxx design or if you failed to procure all the required 74ACxxx parts:



Use this flowchart if you have populated 74ACxxx parts where they are required:



2 Overall Part List

2.1 Detailed list of required parts

2 x	74AC00	IC10, IC63
1 x	74HC00	IC48
1 x	74HC04	IC47
4 x	74AC08	IC16, IC29, IC32, IC62
1 x	74HC08	IC9
1 x	74AC32	IC11
5 x	74AC74	IC14, IC15, IC30, IC31, IC64
5 x	74HC74	IC12, IC13, IC49, IC50, IC34
5 x	74HC138	IC17, IC18, IC19, IC20, IC21
1 x	74AC157	IC56
6 x	74HC161	IC6, IC7, IC35, IC37, IC39, IC41
1 x	74HC245	IC22
8 x	74HC541	IC28, IC36, IC38, IC44, IC46, IC51, IC53, IC65
3 x	74AC574	IC4, IC5, IC8
14 x	74HC574	IC23, IC24, IC25, IC26, IC27, IC40, IC42, IC43, IC45, IC52, IC54, IC55, IC60, IC61
3 x	27C010 or 27C1001, 60ns	IC1, IC2, IC3
1 x	27C64 or 27C256, 100ns	IC59
2 x	27C801 or 27C080, 100ns	IC57, IC58
1 x	LM393	IC33
1 x	10nF ceramic capacitor (no foil !)	C105
67 x	100nF ceramic capacitor (no foil !)	C1 - C65, C100, C101
1 x	2,2µF / 16V, tantalum	C103
2 x	10µF / 16V, tantalum	C102, C104
1 x	10 Ohm	R13
11 x	47 Ohm	R1, R2, R3, R4, R5, R8, R9, R23, R28, R30, R31
1 x	100 Ohm	R18
7 x	470 Ohm	R6, R10, R11, R22, R25, R26, R27
8 x	1 kOhm	R31, R32, R33, R34, R35, R36, R37, R38
1 x	2.2 kOhm	R12
3 x	4.7 kOhm	R19, R21, R29
3 x	10 kOhm	R14, R20, R30
1 x	22 kOhm	R16
1 x	47 kOhm	R15
1 x	220 kOhm	R17
1 x	SIL 8 x 4,7 kOhm	R7
1 x	SIL 8 x 10 kOhm	R24
1 x	D1	1N4148
2 x	Crystal Oscillator	Q1, Q2
4 x	3 pin jumper	JP1, JP2, JP3, JP5
1 x	2*4 pin jumper block + jumper	JP4
1 x	DIN 41612 Connector	J1
1 x	2 pin connector (with cable)	J2
12 x	10 pin header	5 x VA, 5 x VB, 2 x VC
9 x	14 pin header	4 x V1, 3 x V2, 2 x V3
8 x	16 pin header	4 x V0, 2 x V4, 2 x V6
3 x	20 pin header	3 x V5
5 x	printed circuit board, 100x160mm	
16 x	distance husk with screw	

3 Board Stack

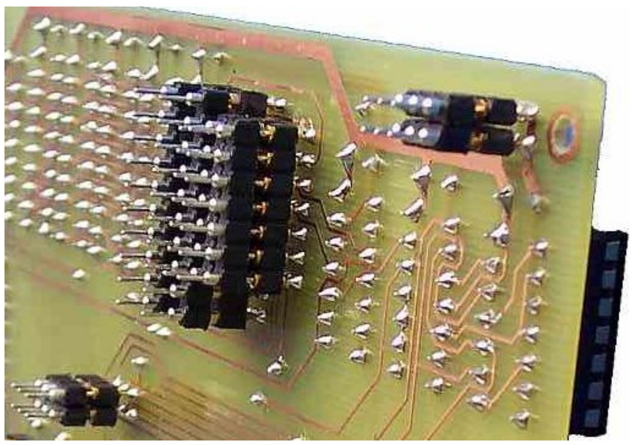
3.1 Stacking the Boards

The CPU consists of 5 printed boards. If wire-wrap connectors with long pins are used for V0 - V6 and VA - VC, the boards can simply put together. The stacking order is as follows:

Fig. 12: MyCPU Board Stack, side view

Note 1:

I am using wire-wrap connectors with long pins. Unfortunately these connectors are no more available (I bought them at www.conrad.com), so you have to find your own alternative. For example you could solder sockets onto the top side, and then solder headers on the bottom side to the same solder pads. The picture at the right shows how I have done it at the memory extension unit (see also the Memory Unit Selfbuild Guide).



Note 2:

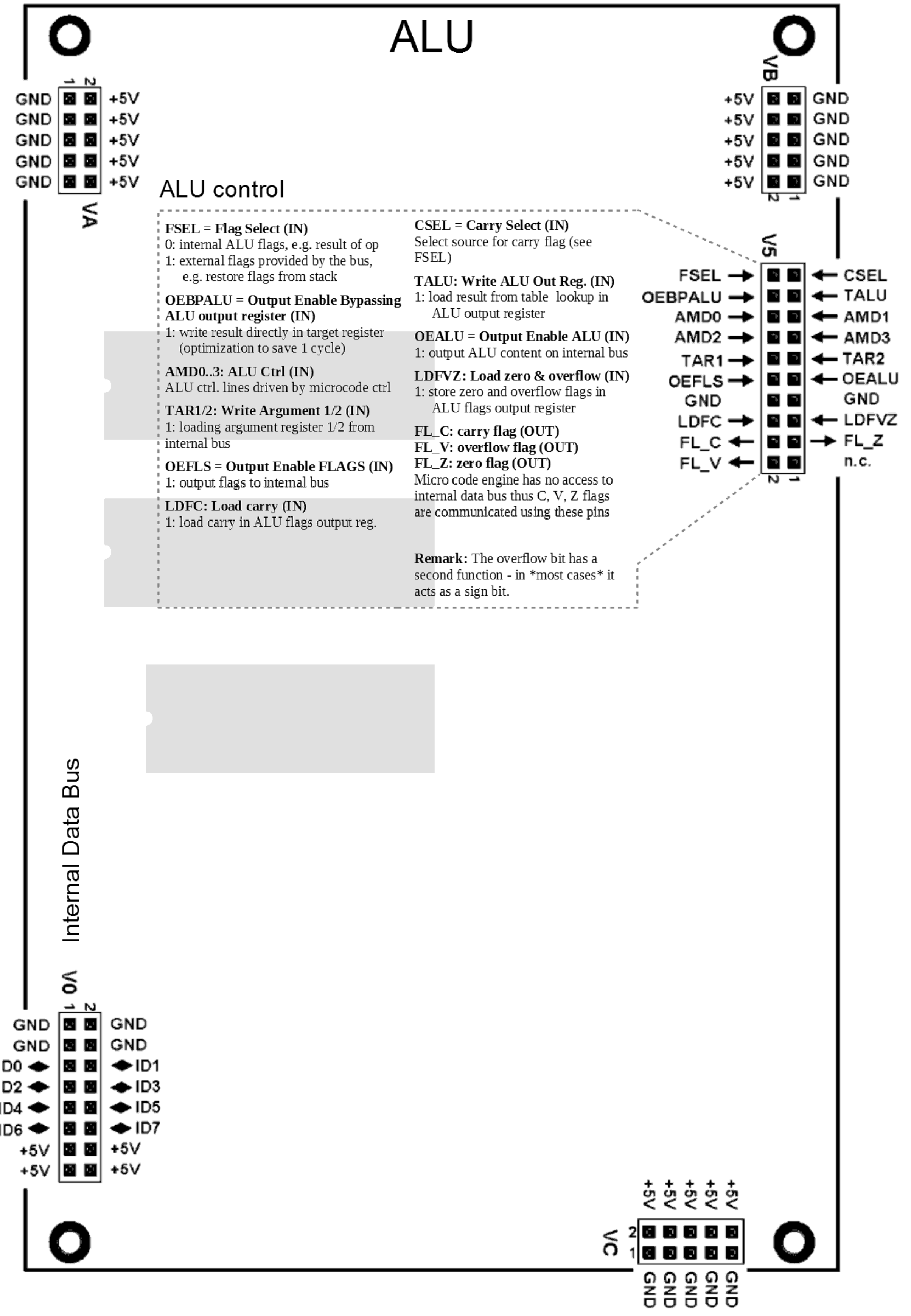
Alternatively you can use ribbon cable with board connectors between the boards. Ribbon cable is much cheaper than the wire-wrap connectors, but it does not look so good, and it must be kept very short to ensure proper CPU operation.

4 Signals on Board Connectors

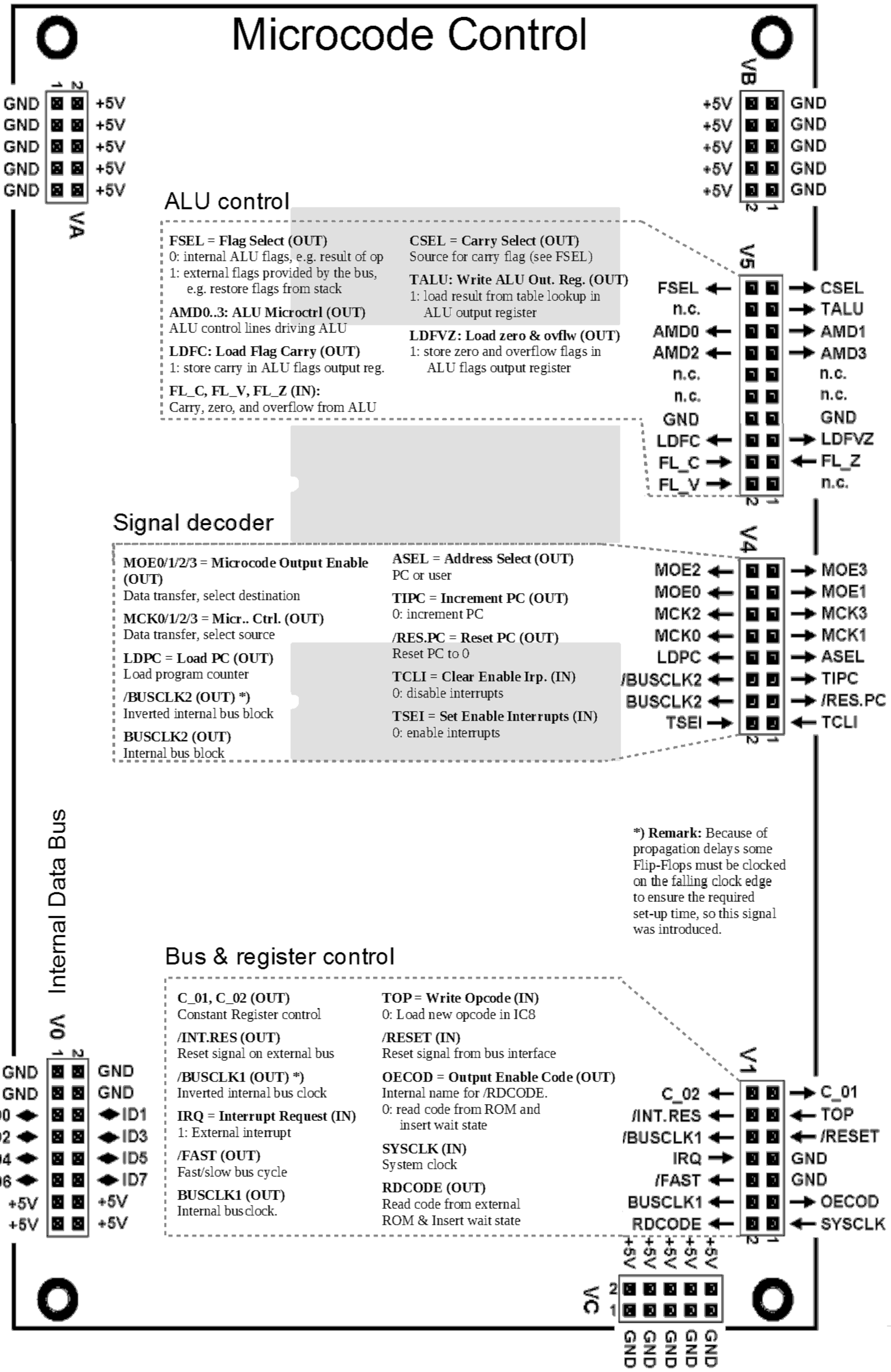
This chapter describes the board connectors. Legend:

- An arrow from the signal name to the connector pin describes an input signal.
- An arrow from the connector pin to the signal name describes an output signal.
- A double-arrow describes a bidirectional signal, in this design it is only the data bus.
- All signal names beginning with a „T“ are edge-triggered, that means the condition „0“ → „1“ triggers the event. These signals are inactive „1“, but are „0“ for one bus clock cycle to generate the rising edge.

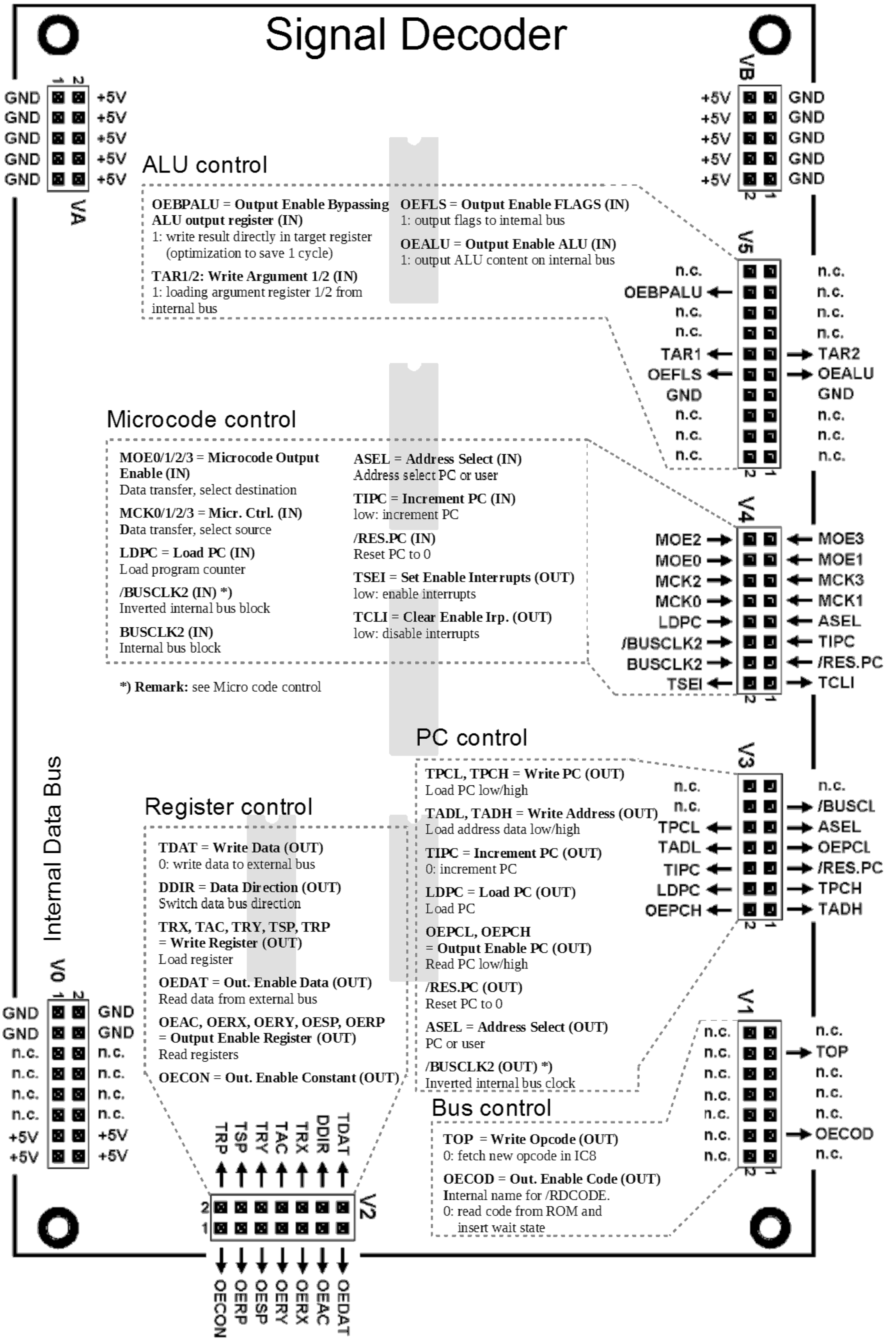
4.1 ALU


ALU
VA
GND +5V
GND +5V
GND +5V
GND +5V
GND +5V
VB
+5V GND
+5V GND
+5V GND
+5V GND
+5V GND
ALU control
FSEL = Flag Select (IN)
0: internal ALU flags, e.g. result of op
1: external flags provided by the bus, e.g. restore flags from stack
OEBPALU = Output Enable Bypassing ALU output register (IN)
1: write result directly in target register (optimization to save 1 cycle)
AMD0..3: ALU Ctrl (IN)
ALU ctrl. lines driven by microcode ctrl
TAR1/2: Write Argument 1/2 (IN)
1: loading argument register 1/2 from internal bus
OEFLS = Output Enable FLAGS (IN)
1: output flags to internal bus
LDFC: Load carry (IN)
1: load carry in ALU flags output reg.
CSEL = Carry Select (IN)
Select source for carry flag (see FSEL)
TALU: Write ALU Out Reg. (IN)
1: load result from table lookup in ALU output register
OEALU = Output Enable ALU (IN)
1: output ALU content on internal bus
LDFVZ: Load zero & overflow (IN)
1: store zero and overflow flags in ALU flags output register
FL_C: carry flag (OUT)
FL_V: overflow flag (OUT)
FL_Z: zero flag (OUT)
Micro code engine has no access to internal data bus thus C, V, Z flags are communicated using these pins
Remark: The overflow bit has a second function - in *most cases* it acts as a sign bit.
V5
FSEL CSEL
OEBPALU TALU
AMD0 AMD1
AMD2 AMD3
TAR1 TAR2
OEFLS OEALU
GND GND
LDFC LDFVZ
FL_C FL_Z
FL_V n.c.
Internal Data Bus
V0
GND GND
GND GND
ID0 ID1
ID2 ID3
ID4 ID5
ID6 ID7
+5V +5V
+5V +5V
VC
+5V +5V +5V +5V +5V
GND GND GND GND GND

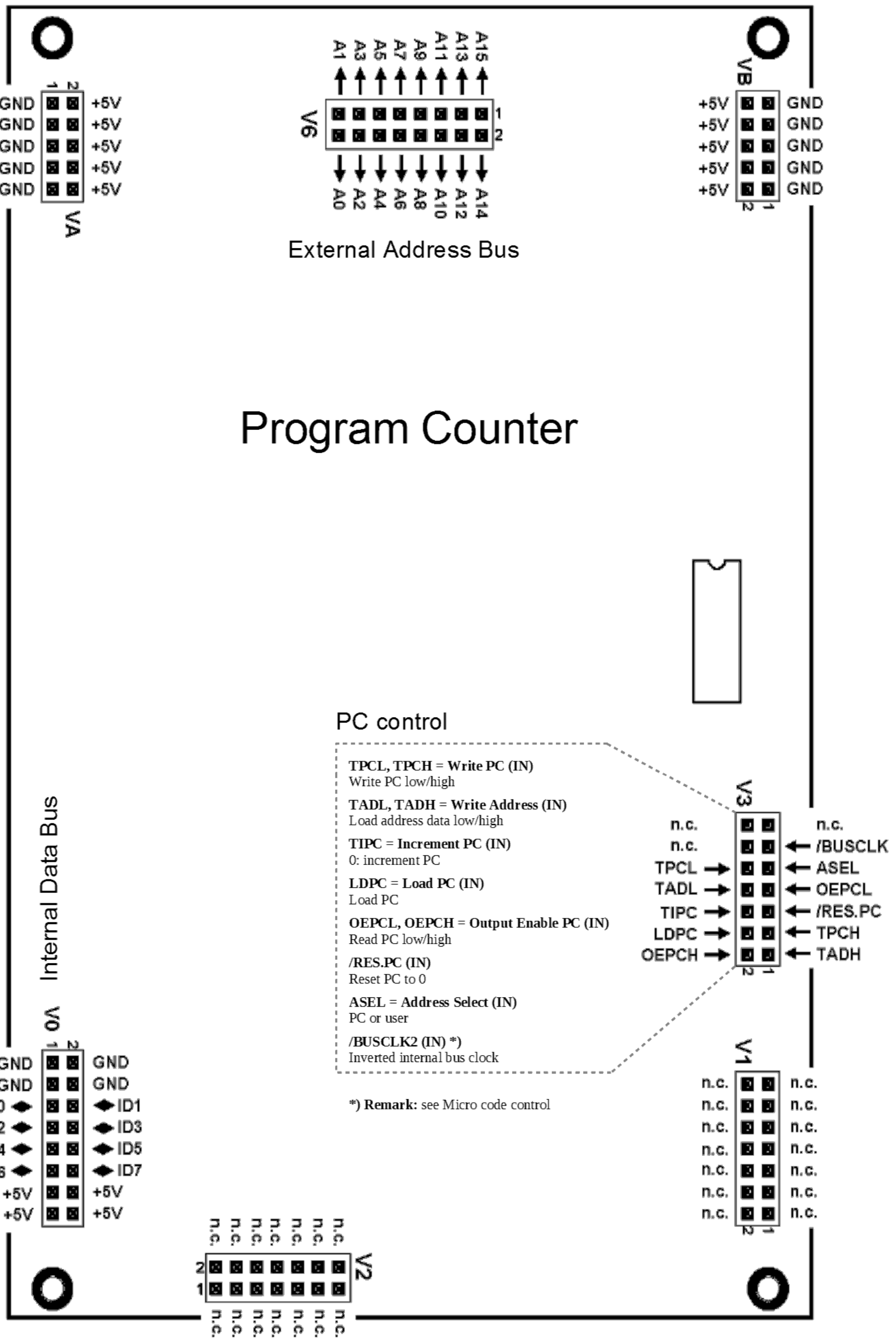
4.2 Control Unit



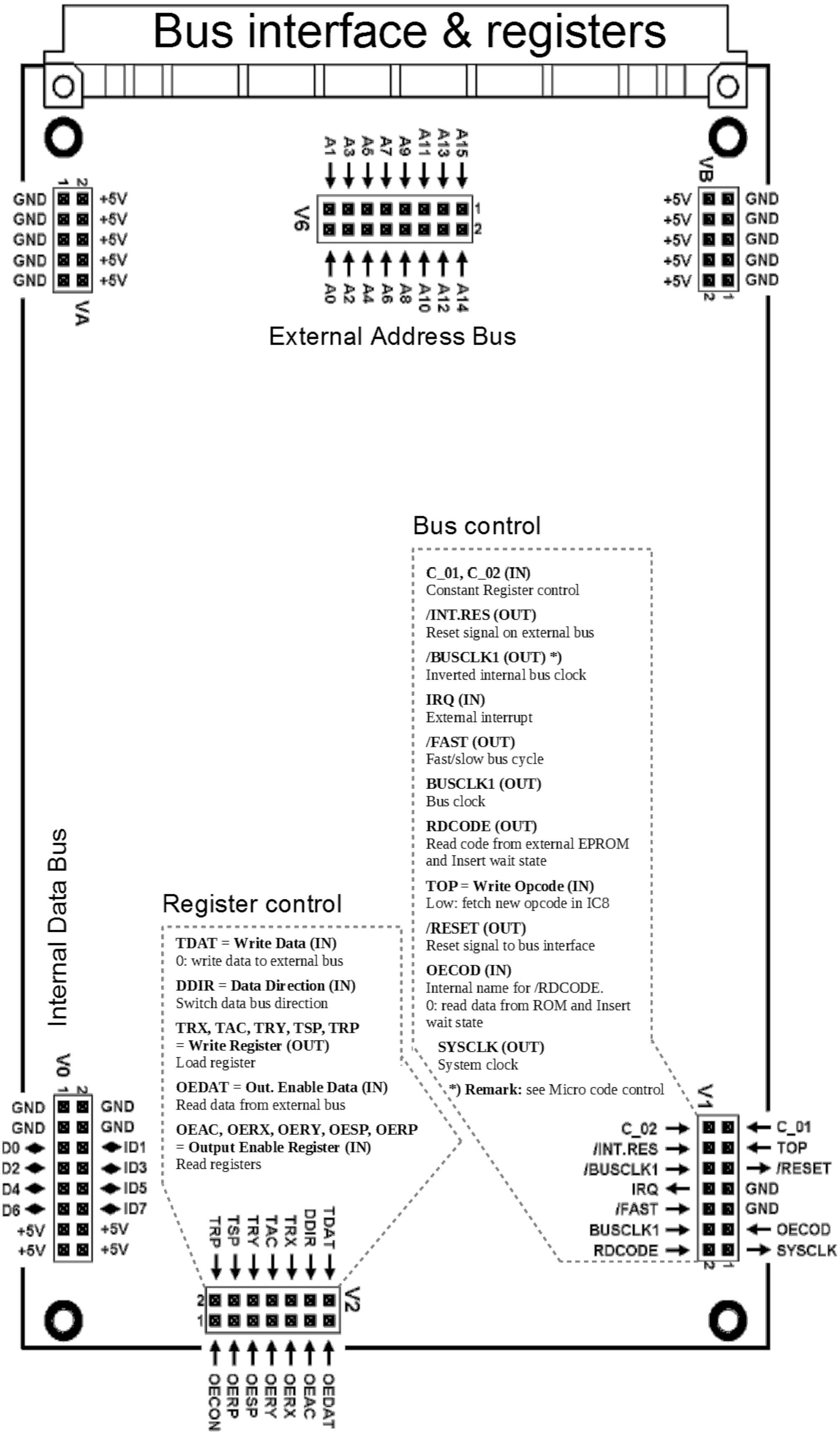
4.3 Decoder


Signal Decoder
ALU control
OEBPALU = Output Enable Bypassing ALU output register (IN)
1: write result directly in target register (optimization to save 1 cycle)
TAR1/2: Write Argument 1/2 (IN)
1: loading argument register 1/2 from internal bus
OEFLS = Output Enable FLAGS (IN)
1: output flags to internal bus
OEALU = Output Enable ALU (IN)
1: output ALU content on internal bus
Microcode control
MOE0/1/2/3 = Microcode Output Enable (IN)
Data transfer, select destination
MCK0/1/2/3 = Micr. Ctrl. (IN)
Data transfer, select source
LDPC = Load PC (IN)
Load program counter
/BUSCLK2 (IN) *)
Inverted internal bus block
BUSCLK2 (IN)
Internal bus block
ASEL = Address Select (IN)
Address select PC or user
TIPC = Increment PC (IN)
low: increment PC
/RES.PC (IN)
Reset PC to 0
TSEI = Set Enable Interrupts (OUT)
low: enable interrupts
TCLI = Clear Enable Irp. (OUT)
low: disable interrupts
*) Remark: see Micro code control
PC control
TPCL, TPCH = Write PC (OUT)
Load PC low/high
TADL, TADH = Write Address (OUT)
Load address data low/high
TIPC = Increment PC (OUT)
0: increment PC
LDPC = Load PC (OUT)
Load PC
OEPCL, OEPCH = Output Enable PC (OUT)
Read PC low/high
/RES.PC (OUT)
Reset PC to 0
ASEL = Address Select (OUT)
PC or user
/BUSCLK2 (OUT) *)
Inverted internal bus clock
Register control
TDAT = Write Data (OUT)
0: write data to external bus
DDIR = Data Direction (OUT)
Switch data bus direction
TRX, TAC, TRY, TSP, TRP = Write Register (OUT)
Load register
OEDAT = Out. Enable Data (OUT)
Read data from external bus
OEAC, OERX, OERY, OESP, OERP = Output Enable Register (OUT)
Read registers
OECON = Out. Enable Constant (OUT)
Bus control
TOP = Write Opcode (OUT)
0: fetch new opcode in IC8
OECOD = Out. Enable Code (OUT)
Internal name for /RDCODE.
0: read code from ROM and insert wait state
Internal Data Bus
VA
VB
V5
V4
V3
V2
V1
V0
GND
+5V
n.c.
OEBPALU
TAR1
TAR2
OEFLS
OEALU
MOE2
MOE3
MOE0
MOE1
MCK2
MCK3
MCK0
MCK1
LDPC
ASEL
/BUSCLK2
TIPC
BUSCLK2
/RES.PC
TSEI
TCLI
/BUSCL
TPCL
TADL
OEPCL
TIPC
/RES.PC
LDPC
TPCH
OEPCH
TADH
TOP
OECOD
TRP
TSP
TRY
TAC
TRX
DDIR
TDAT
OECON
OERP
OESP
OERY
OERX
OEAC
OEDAT

4.4 Program Counter



4.5 Interface Board



5 CPU Bus Connector

5.1 Layout of the Bus Connector

CPU Bus Connector (DIN41612 / IEC603-2)

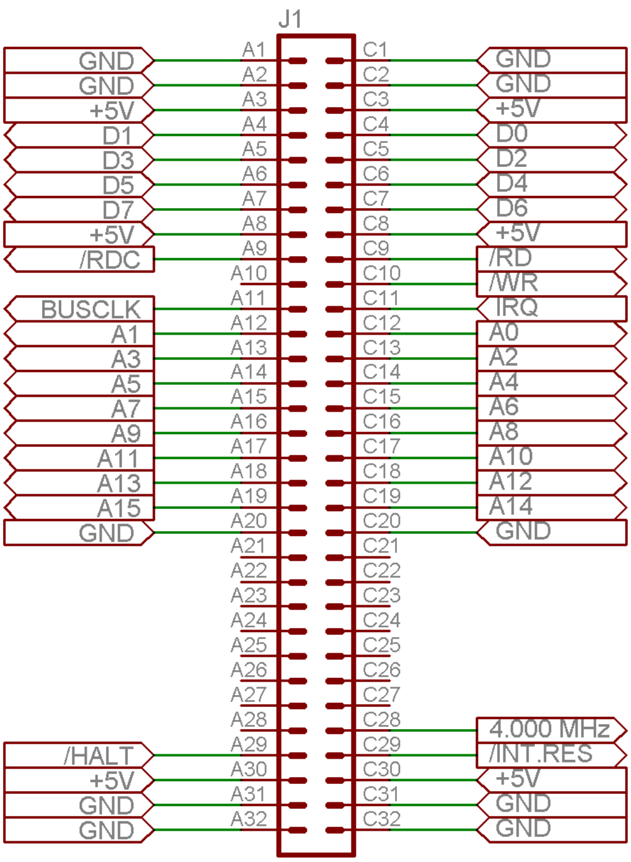


Fig. 13: CPU Bus Connector

A1	GND	C1	GND
A2	GND	C2	GND
A3	+5V	C3	+5V
A4	D1	C4	D0
A5	D3	C5	D2
A6	D5	C6	D4
A7	D7	C7	D6
A8	+5V	C8	+5V
A9	/RDC(code)	C9	/RD (data)
A10	reserved	C10	/WR (data)
A11	BUSCLK	C11	IRQ
A12	A1	C12	A0
A13	A3	C13	A2
A14	A5	C14	A4
A15	A7	C15	A6
A16	A9	C16	A8
A17	A11	C17	A10
A18	A13	C18	A12
A19	A15	C19	A14
A20	GND	C20	GND
A21	(+12V)	C21	(+12V)
A22	(-12V)	C22	(-12V)
A23	(IRQ1)	C23	(IRQ0)
A24	(IRQ3)	C24	(IRQ2)
A25	(IRQ5)	C25	(IRQ4)
A26	(IRQ7)	C26	(IRQ6)
A27	(/IOEN2)	C27	(/IOEN1)
A28	(/IOEN3)	C28	4.000MHz
A29	/HALT	C29	/INT.RES
A30	+5V	C30	+5V
A31	GND	C31	GND
A31	GND	C32	GND

5.2 Bus Signal Description

Pin	Direction	Active Level	Description
A1			GND
A2			GND
A3			+5V
A4	bidirectional	high	D1 Bidirectional data bus line
A5	bidirectional	high	D3 Bidirectional data bus line
A6	bidirectional	high	D5 Bidirectional data bus line
A7	bidirectional	high	D7 Bidirectional data bus line
A8			+5V
A9	output	low	/RDC Read code memory (max. 64 kByte). If this line is low, the CPU

			reads data from the code memory (ROM). The address lines are valid at least one half CPU clock cycle before this line is asserted. The exact timing of this signal depends on the CPU microcode.
A10			Reserved
A11	output	high	BUSCLK Bus clock, has the same frequency like CPU core frequency and is phase synchronous to the /RDC, /RD and /WR lines.
A12	output	high	A1 Address bus line
A13	output	high	A3 Address bus line
A14	output	high	A5 Address bus line
A15	output	high	A7 Address bus line
A16	output	high	A9 Address bus line
A17	output	high	A11 Address bus line
A18	output	high	A13 Address bus line
A19	output	high	A15 Address bus line
A20			GND
A21			(reserved for +12V supply on backplane)
A22			(reserved for -12V supply on backplane)
A23			(reserved for IRQ1 input line to interrupt controller board)
A24			(reserved for IRQ3 input line to interrupt controller board)
A25			(reserved for IRQ5 input line to interrupt controller board)
A26			(reserved for IRQ7 input line to interrupt controller board)
A27			(reserved for /IOEN2 output line of memory board)
A28			(reserved for /IOEN3 output line of memory board)
A29	input	low	/HALT Input with internal pull-up. If asserted the CPU is halted (the internal clock is switched off). When the CPU is halted, the signal BUSCLK is low, but the 4.000MHz-line is still working. The /HALT-signal can be used by slow I/O-devices to lengthen the current I/O-access.
A30			+5V
A31			GND
A31			GND
C1			GND
C2			GND
C3			+5V
C4	bidirectional	high	D0 Bidirectional data bus line
C5	bidirectional	high	D2 Bidirectional data bus line
C6	bidirectional	high	D4 Bidirectional data bus line
C7	bidirectional	high	D6 Bidirectional data bus line
C8			+5V
C9	output	low	/RD Read data memory (max. 64 kByte). If this line is low, the CPU reads data from the data memory (RAM) or an I/O-device. The address lines are valid at least one half CPU clock cycle before this line is asserted. The exact timing of this signal depends on the CPU microcode.

C10	output	low	/WR Write data memory (max. 64 kByte). If this line is low, the CPU writes data to the data memory (RAM) or to an I/O-device. The address lines are valid at least one half CPU clock cycle before this line is asserted. The exact timing of this signal depends on the CPU microcode.
C11	input	high	IRQ Input with internal pull-down. Maskable Interrupt Input. If this line is asserted, the CPU leaves the normal program execution and executes an interrupt service routine.
C12	output	high	A0 Address bus line
C13	output	high	A2 Address bus line
C14	output	high	A4 Address bus line
C15	output	high	A6 Address bus line
C16	output	high	A8 Address bus line
C17	output	high	A10 Address bus line
C18	output	high	A12 Address bus line
C19	output	high	A14 Address bus line
C20			GND
C21			(reserved for +12V supply on backplane)
C22			(reserved for -12V supply on backplane)
C23			(reserved for IRQ0 input line to interrupt controller board)
C24			(reserved for IRQ2 input line to interrupt controller board)
C25			(reserved for IRQ4 input line to interrupt controller board)
C26			(reserved for IRQ6 input line to interrupt controller board)
C27			(reserved for /IOEN1 output line of memory board)
C28	output	high	4.000 MHz Clock This clock is not synchronous to the CPU core frequency. It remains active when the /HALT-signal is asserted. This clock can be used by hardware devices for bus delay timing.
C29	output	low	/INT.RES Synchronous reset output. On power-on, this line is low for several 100 milliseconds. It can be used to reset external hardware devices.
C30			+5V
C31			GND
C32			GND

Tab. 1: CPU Bus Signals

6 CPU Reset And Interrupt Sequence

6.1 Reset Behaviour

On power-up, IC33 generates a reset pulse with a duration of 100ms. This is enough time for the crystal oscillators to start swinging. The CPU logic needs 3 core clock cycles (= 12-16 crystal clocks) for the reset to take effect. That means, when the reset is asserted, the crystal has to swing 12-16 times until all the internal CPU logic is initialized. While the reset signal is asserted, the CPU fetches the OP-code at address 0000h again and again.

After IC33 has finished the reset the CPU executes the OP-code fetched from address 0000h. In most cases a jump command is stored at address 0000h that points to the reset routine in the ROM.

After the reset the interrupt is masked (disabled), and all registers have undefined values.

6.2 Hardware Interrupt

The hardware interrupt is microcode-dependent. With the MyCPU23-microcode the CPU executes the code at address 0003h when the interrupt line is asserted (= set high).

But before the CPU starts executing the code at address 0003h, the current program counter and the flags register are saved to the stack. Also all further interrupts are disabled by masking the interrupt line. When the CPU executes the RTI (=return from interrupt) -instruction, the interrupt line is automatically enabled again, the flags are restored from stack and the processor continues execution of the main program.

Note: If the interrupt line is asserted while the CPU executes the RTI command, the CPU will immediately jump to address 0003h again, without executing any OP-code from the main program stream.

6.3 Software Interrupt

The MyCPU23-microcode provides one software interrupt. The software interrupt is executed when the CPU fetches a BRK (=break program flow) -instruction. The CPU then continues program execution at ROM address 0006h. At this point the CPU has stored the old program counter and the flags to stack memory to be able to return to the interrupted program after the software interrupt has been serviced. The difference to the hardware interrupt is that the hardware interrupt line is not automatically masked when the software interrupt is raised. Because of this there exists a different instruction for finishing the software interrupt: RTB (=return from break interrupt).

Address (Interrupt Vector)	Type of interrupt
0000h	Hardware Reset
0003h	Hardware Interrupt
0006h	Software Interrupt

Tab. 2: CPU Interrupt Vectors

7 Backplane

7.1 Backplane Cable

The backplane consists of 0.35 meters of a 64 wire ribbon cable and 9 DIN 41612 connectors:

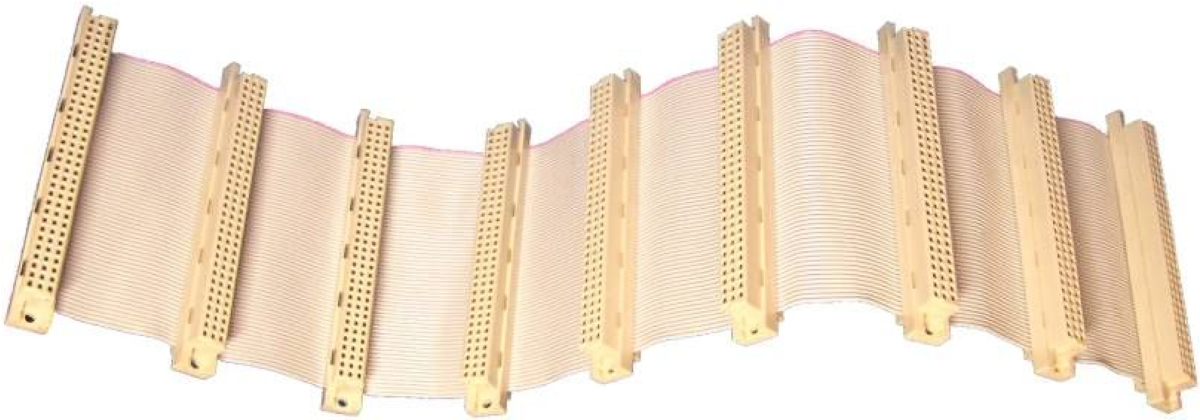


Fig. 14: MyCPU Backplane Cable

8 Schematics

8.1 List of all Schematics

Fig. 15:	ALU, Control and Registers
Fig. 16:	ALU, Look-up Tables
Fig. 17:	Microcode and common steering
Fig. 18:	Program Counter, lower 8 address lines
Fig. 19:	Program Counter, upper 8 address lines
Fig. 20:	General Purpose Registers
Fig. 21:	Clock-generation and –control
Fig. 22:	Reset Generator
Fig. 23:	Bus Interface

8.2 Schematics sorted by Boards

Board 1, Arithmetic Logic Unit	Fig. 15, Fig. 16
Board 2, Microcode Control	Fig. 17
Board 3, Signal Decode	Fig. 17
Board 4, Program Counter	Fig. 18, Fig. 19
Board 5, Register and Interface	Fig. 20, Fig. 21, Fig. 22, Fig. 23

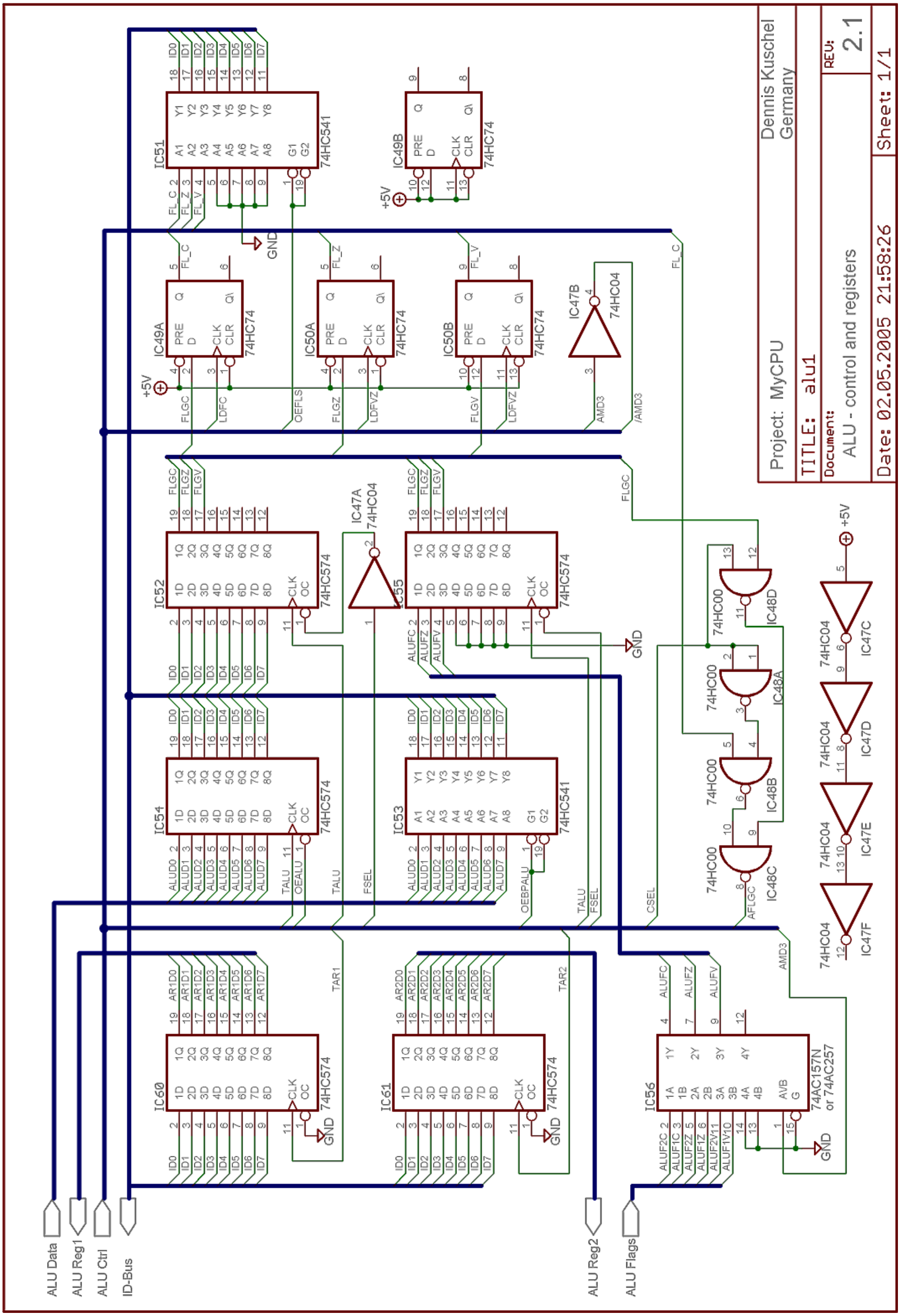


Fig. 15: ALU, Control and Registers

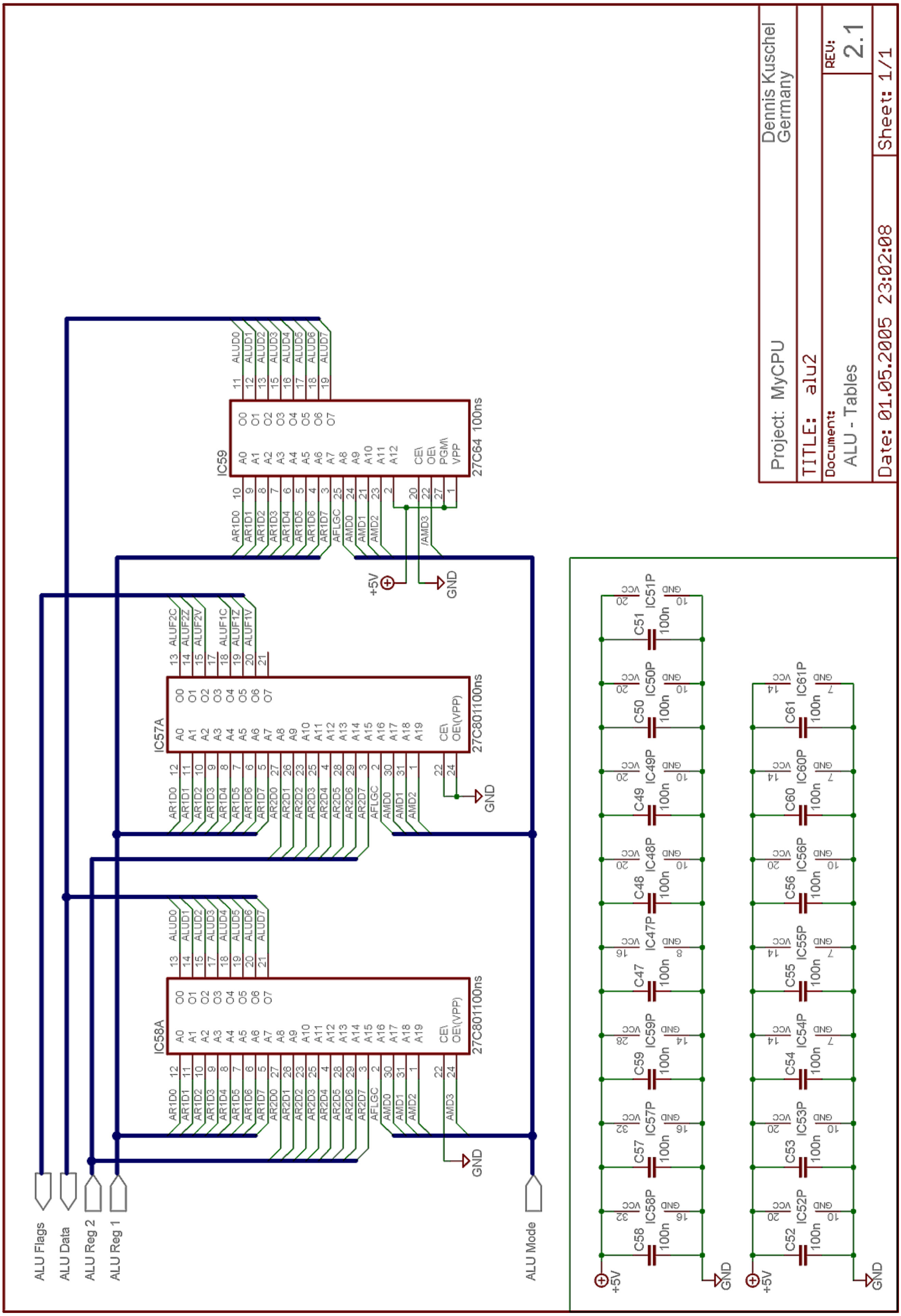


Fig. 16: ALU, Look-up Tables

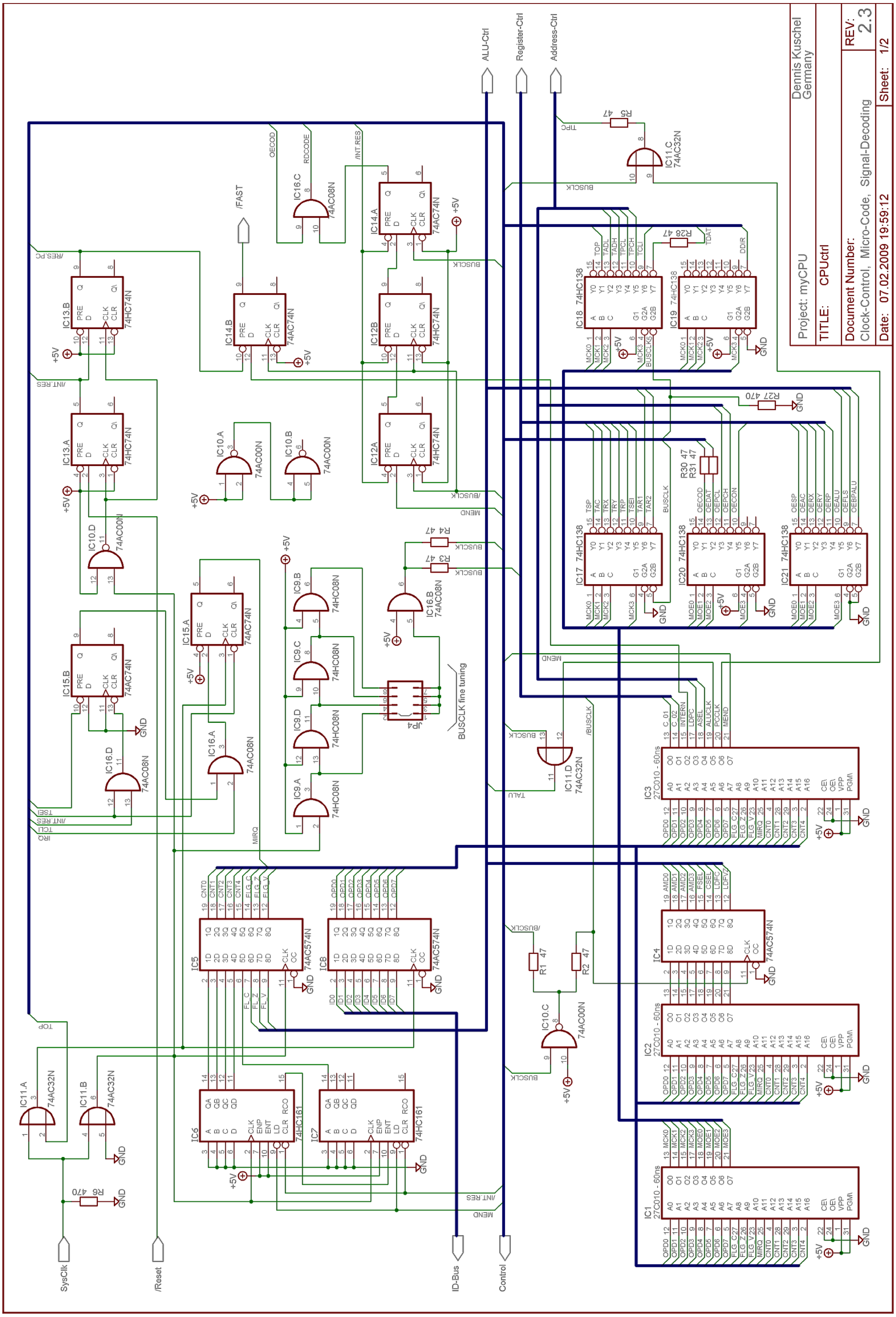


Fig. 17: Microcode and common steering

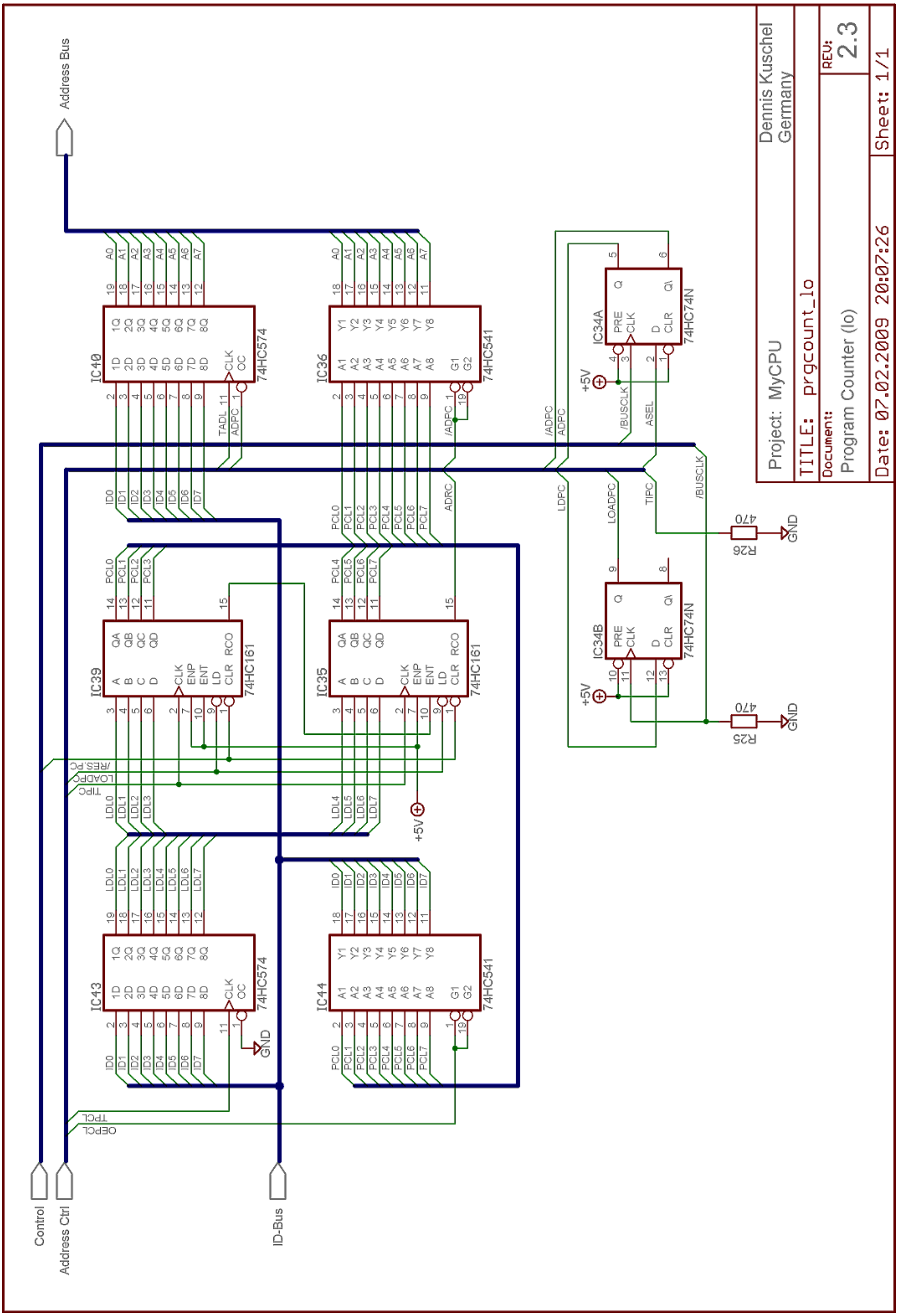


Fig. 18: Program Counter, lower 8 address lines

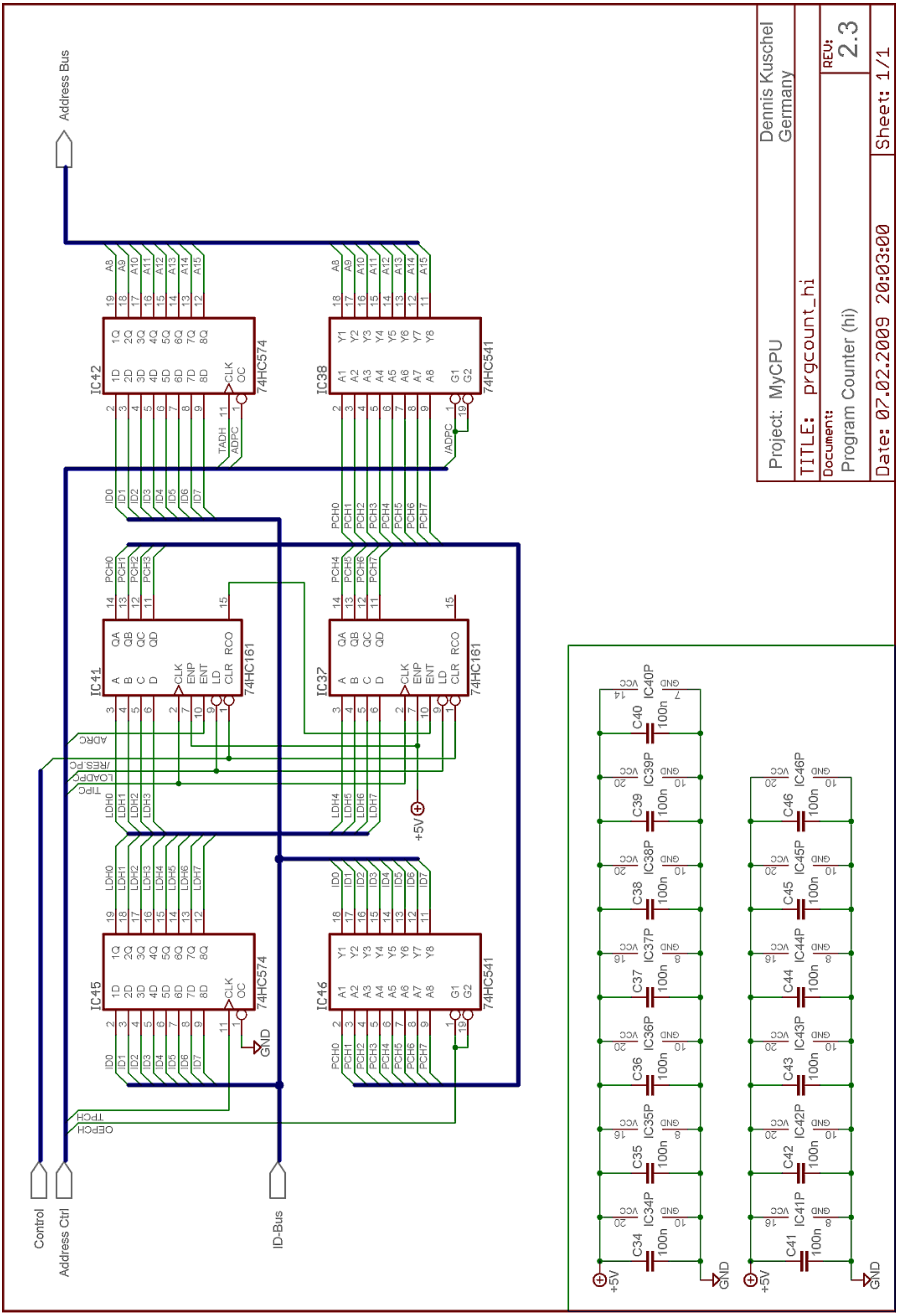


Fig. 19: Program Counter, upper 8 address lines

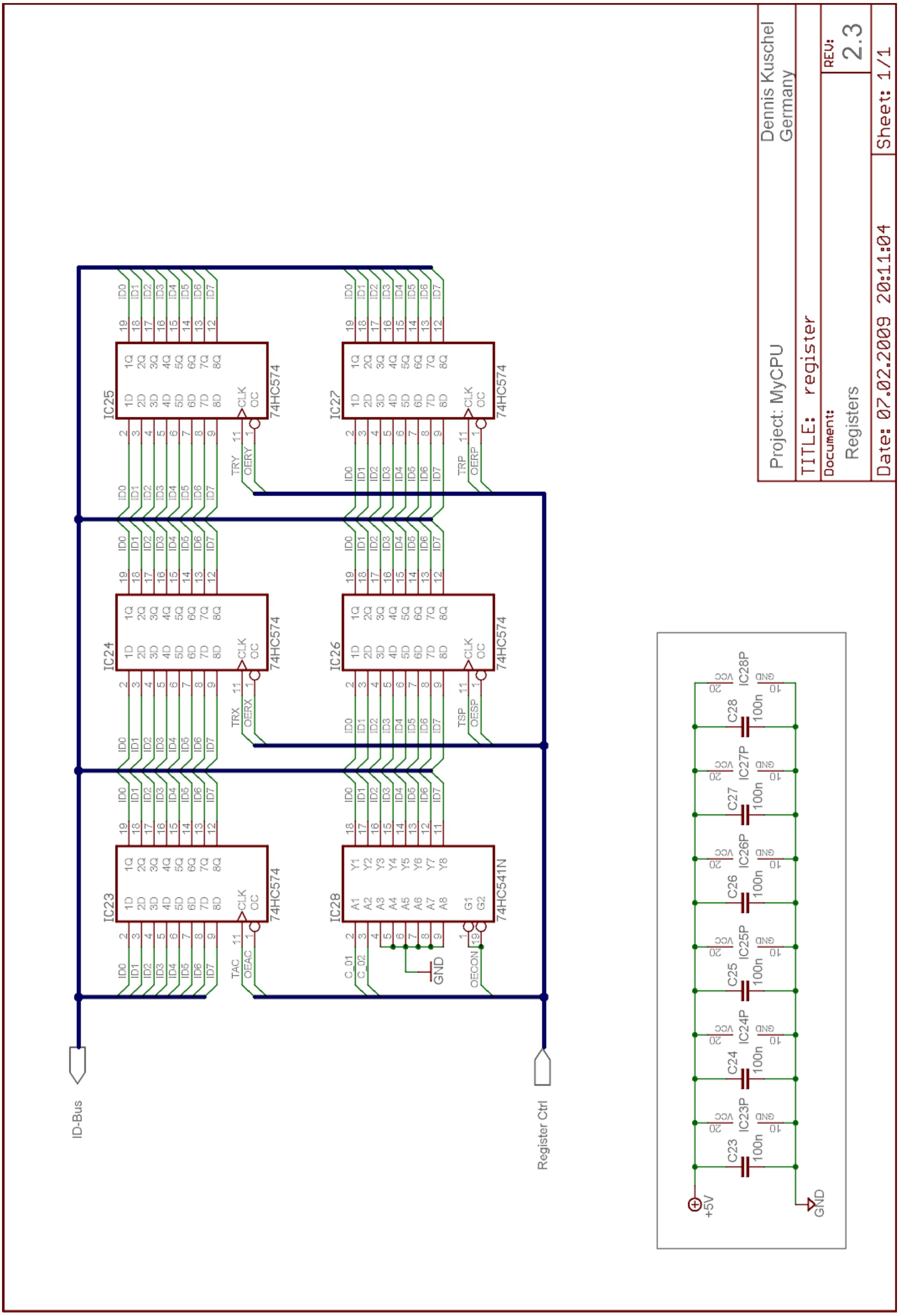


Fig. 20: General Purpose Registers

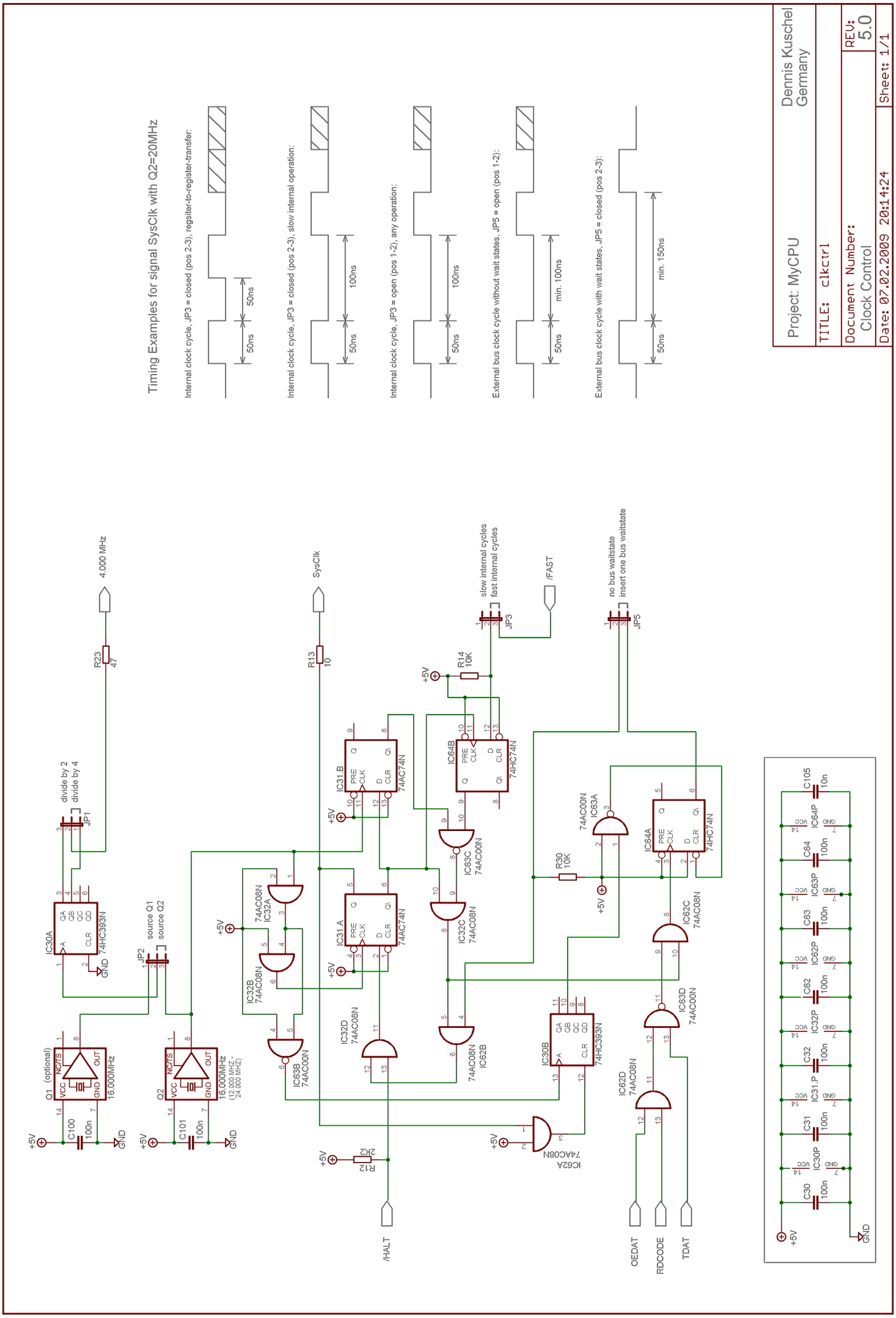


Fig. 21: Clock-generation and –control

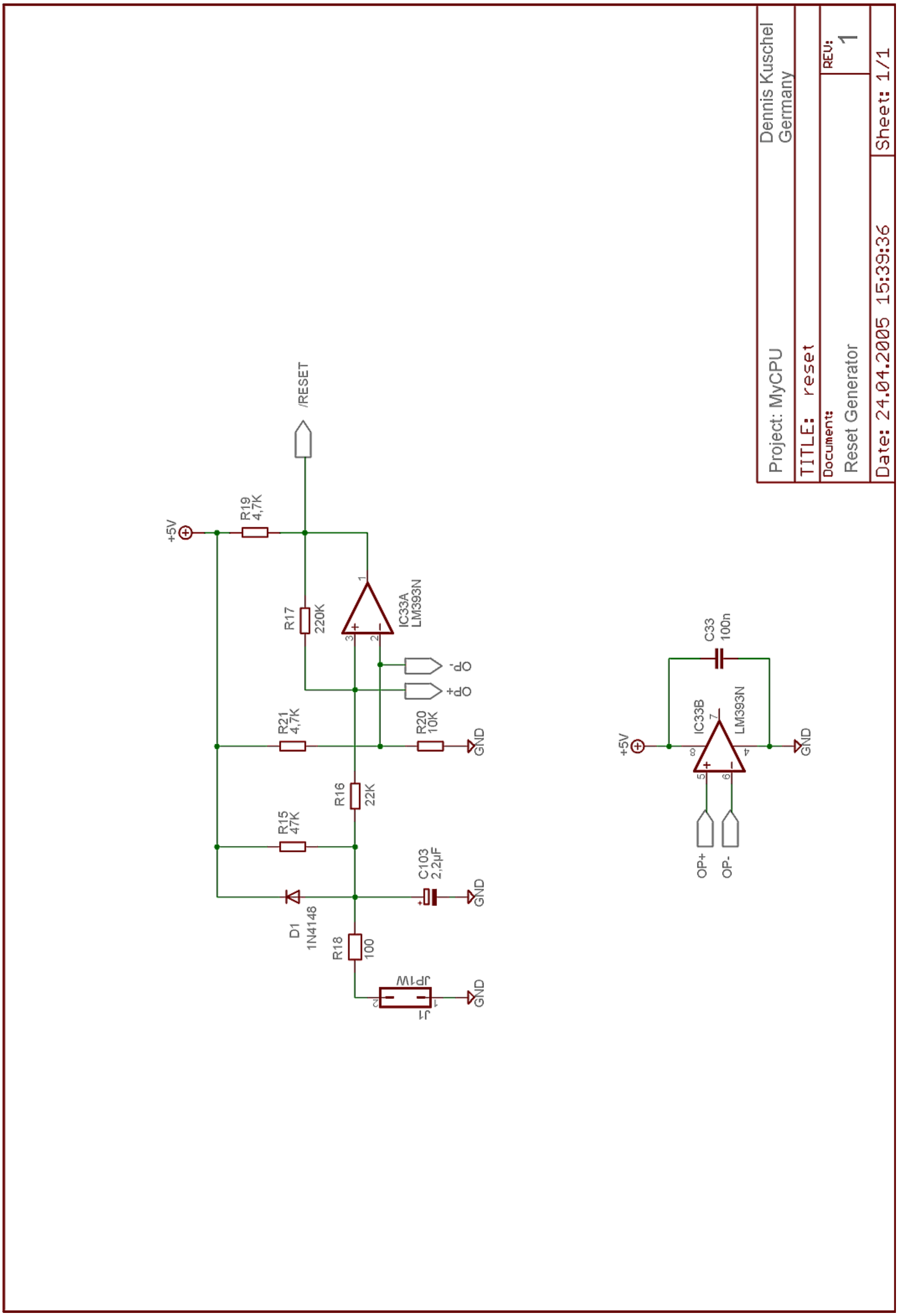


Fig. 22: Reset Generator

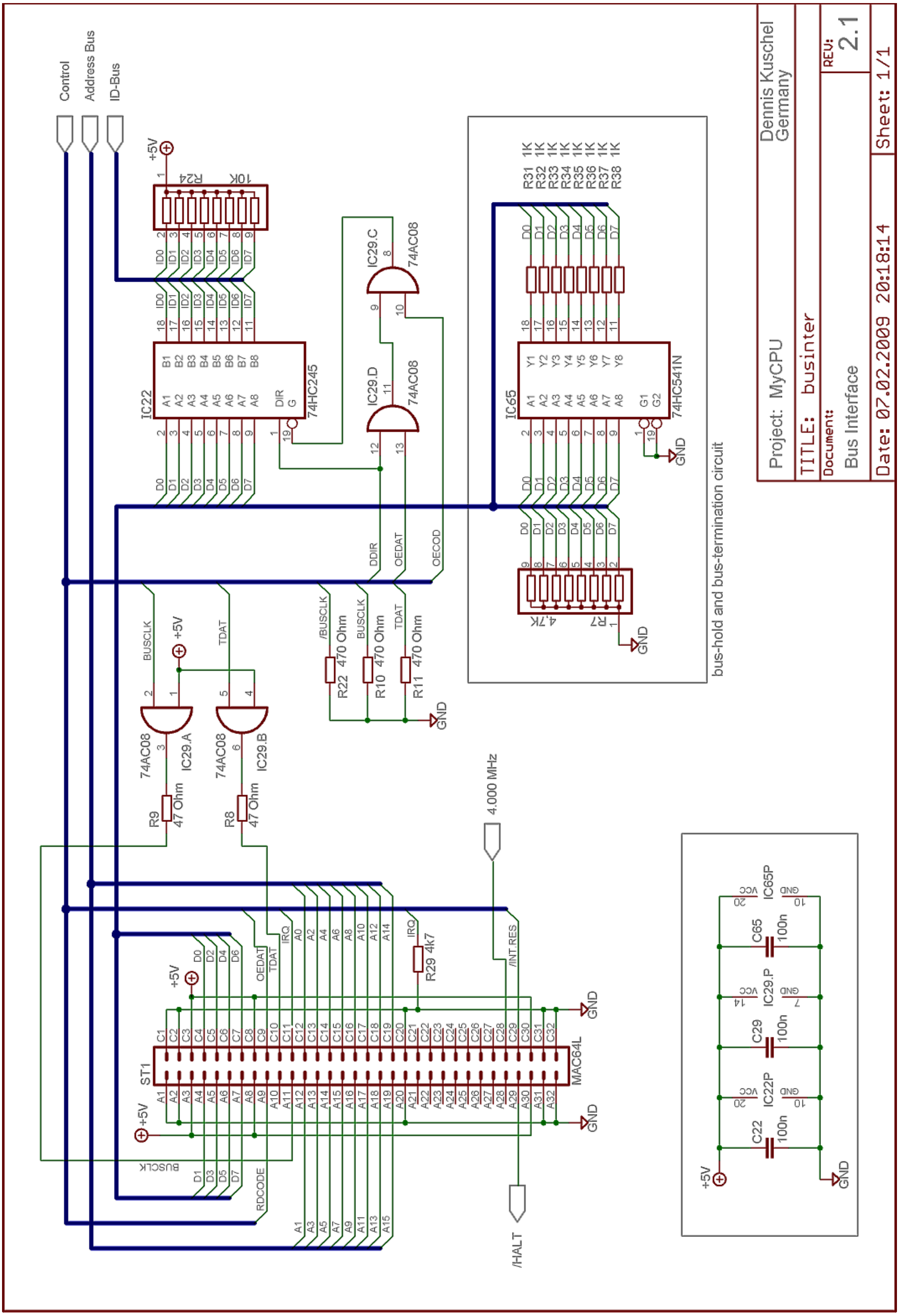


Fig. 23: Bus Interface

9 Change Log

9.1 Changes in the MyCPU design

Date	Name	Chapter	Description
2006-12-16	D.Kuschel	1.4	IC36, IC38 replaced by 74HC541 IC40, IC42 replaced by 74HC574
			Information on page IV inserted.
		4	Chapter “Signal on Board Connectors” added
2007-01-13	D.Kuschel	1.5	Interface Board replaced by new version, chapter reworked.
		8	Schematics of Bus-Interface and Clock-Control updated
		2.1	Table updated
2008-09-24	D.Kuschel	1.5	Layout of Interface Board changed (bug fixed)
		1.5.3	Values of R8 and R13 changed to 10 Ohm
		8.2	Figure 21 and 23 updated
2009-02-08	D.Kuschel	all	Updated to MyCPU v2.3
2015-07-23	D.Kuschel	3, 4	Chapters revised
2015-07-23	D.Kuschel		Added a warning about the use of film capacitors