Instruction Set Architecture (ISA)

Instruction Set

Instruction	Type	Operation	Binary Encoding

ADD	RRR	regA = regB + regC
		F = 0/1 means unsigned/ signed addition
			0000 regA regB regC

SUB	RRR	regA = regB - regC
		F = 0/1 means unsigned/ signed subtraction
			0001 regA regB regC

OR	RRR	regA = regB OR regC	0010 regA regB regC
AND	RRR	regA = regB AND regC	0011 regA regB regC
LI	RI	load IM into regA F = 0/1 means 8bit immediate value gets put into the upper/ lower 16bit of regA with 0x00 filling the remaining bits	0100 regA IM
SW	RRS	store word of regC into address of regB	0101 regB regC
LW	RRL	load word at address of regB into regA	0110 regA regB
BEQ	RRS	branch to address of regB if regC == CMP-bit F-Bit and 2 unused bits (bit 8 + 1 + 0) are used to decide if equal using one of the CMP_BITs from CMP command	0110 regB regC
B	I	branch to immediate address	0111 IM
SLR	RRR	regA = regB >> regC	1000 regA regB regC
SLL	RRR	regA = regB << regC	1001 regA regB regC
CMP	RRR	compare regB == regC and write to regA sets one of the following bits in regA based on comparison of regB and regC for further evaluation with BEQ: - CMP_BIT_EQ (bit 14 MSB) - CMP_BIT_BGC (bit 13 MSB) - CMP_BIT_BLC (bit 12 MSB) - CMP_BIT_BZ (bit 11 MSB) - CMP_BIT_CZ (bit 10 MSB)	1010 regA regB regC
NOP	-	No operation	-
HALT	-	Halt execution	-

Table 1: ISA Instructions

Instruction Format Details

<table>
<tr><td>Bit:</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td></td><td colspan="4">4 bit</td><td colspan="3">3 bit</td><td>1 bit</td><td colspan="3">3 bit</td><td colspan="3">3 bit</td><td colspan="2">2 bit</td></tr>
<tr><td>RRR</td><td colspan="4">opcode</td><td colspan="3">reg A</td><td>F</td><td colspan="3">regB</td><td colspan="3">reg C</td><td colspan="2">unused</td></tr>
<tr><td></td><td colspan="4">4 bit</td><td colspan="3">3 bit</td><td>1 bit</td><td colspan="3">3 bit</td><td colspan="5">5 bit</td></tr>
<tr><td>RRL</td><td colspan="4">opcode</td><td colspan="3">reg A</td><td>F</td><td colspan="3">reg B</td><td colspan="5">unused</td></tr>
<tr><td></td><td colspan="4">4 bit</td><td colspan="3">3 bit</td><td>1 bit</td><td colspan="3">3 bit</td><td colspan="3">3 bit</td><td colspan="2">2 bit</td></tr>
<tr><td>RRS</td><td colspan="4">opcode</td><td colspan="3">unused</td><td>F</td><td colspan="3">reg B</td><td colspan="3">reg C</td><td colspan="2">unused</td></tr>
<tr><td></td><td colspan="4">4 bit</td><td colspan="3">3 bit</td><td>1 bit</td><td colspan="8">8 bit</td></tr>
<tr><td>RI</td><td colspan="4">opcode</td><td colspan="3">reg A</td><td>F</td><td colspan="8">immediate (0 to 0xFF)</td></tr>
<tr><td></td><td colspan="4">4 bit</td><td colspan="3">3 bit</td><td>1 bit</td><td colspan="8">8 bit</td></tr>
<tr><td>I</td><td colspan="4">opcode</td><td colspan="3">unused</td><td>F</td><td colspan="8">immediate (0 to 0xFF)</td></tr>
</table>

Schaltbild

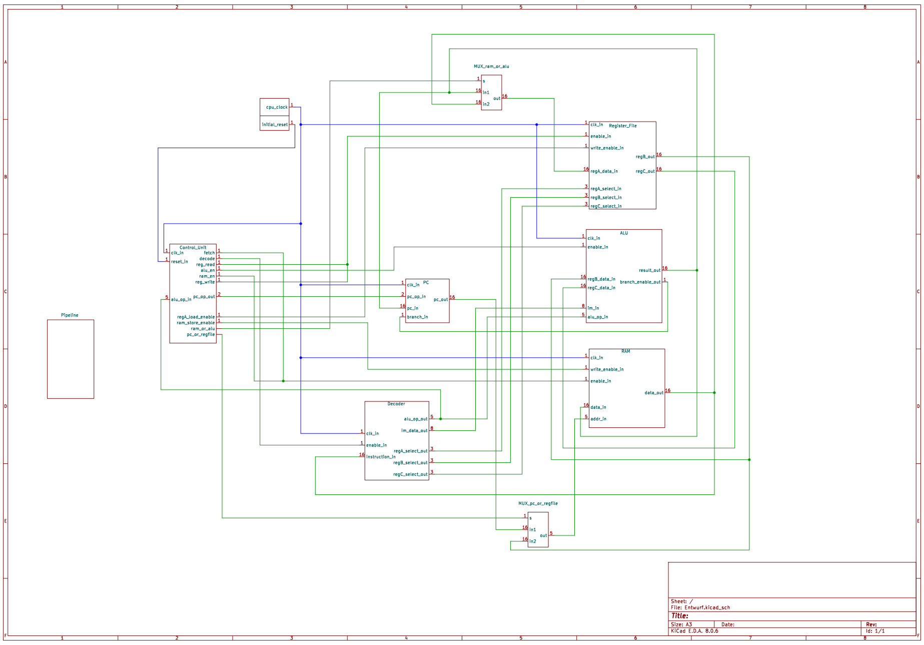

Figure 1: Schaltbild