EDUC–8L: Adding a Link Bit to the EDUC–8 Microcomputer

by
Steven S. Pietrobon, Ph.D.
27 October 2024.

Abstract

The Electronics Australia EDUC–8 microcomputer from 1974 is an 8–bit version of the 12–bit Digital PDP–8. In order to reduce complexity, the PDP–8 link bit was not included in EDUC–8. This paper describes the changes required to EDUC–8 in order to add the link bit, along with 16 new instructions. The changes add 12 additional 74 series TTL chips to the original 100 chips in EDUC–8.

Introduction

The link bit is an extra register in the PDP–8 that is very useful when performing additions and multiplies where the operands have two or more words. The basic operation is that the link bit L is inverted when the carry–out from a TAD (two's addition) or IAC (increment accumulator) operation is equal to one. In the EDUC–8 the RAL and RAR instructions do not include the link bit. We have that

RAL Rotate Accumulator Left (AC[7:0] := AC[6:0],AC[7]),
RAR Rotate Accumulator Right (AC[7:0] := AC[0],AC[7:1].

In the PDP–8, RAL and RAR include the link bit in their rotations. In order to allow upward compatibility with EDUC–8, we use instruction names RLL and RLR. We have that

RLL Rotate Link Left (L := AC[7], AC[7:0] := AC[6:0],L),
RLR Rotate Link Right (L := AC[0], AC[7:0] := L,AC[7:1]).

The PDP–8 includes the following instructions, which we also include in EDUC–8L.

CLL Clear Link (L := 0),
CML Complement Link (L := $\overline{L}$),
SZL Skip on Zero Link (skip the next instruction if L = 0)
SNL Skip on Non–zero Link (skip the next instruction if L = 1).

In our upgrade, we also provide two combined instructions.

STL = CLL.CML Set Link (L := 1),
SKP = SZL.SNL Skip (skip the next instruction).

In order to implement the link bit, changes to the E8/T (timing), E8/D (decoder), E8/A (accumulator), E8/P (program counter), E8/C (connector) and E8/F (front panel) boards are required. The new boards are called E8L/T, E8L/D, E8L/A, E8L/P, E8L/C and E8L/F, respectively. The E8L/C board also includes changes required for the EDUC–8ME memory and I/O port extension [1].

The PDP–8 also includes other instructions that are were not implemented in EDUC–8 in order to reduce its complexity. We also include modifications for these instructions, as well as one instruction that was not implemented on the PDP–8, namely DAC (decrement accumulator). The list of these new instructions are below. For the skip instructions, AC is assumed to be a two's complement value with AC[7] indicating the sign bit, 0 for positive (0 to 127) and 1 for negative (–128 to –1).

DAC Decrement Accumulator (AC := AC–1),
OSR OR Switch Register (AC := AC or SR),
BSW Byte Swap (AC[7:0] := AC[3:0],AC[7:4]),
RTL Rotate Twice Left (L := AC[6], AC[7:0] := AC[5:0],L,AC[7]),
RTR Rotate Twice Right (L := AC[1], AC[7:0] := AC[0],L,AC[6:2]),
SNA Skip on Non–Zero Accumulator (skip the next instruction if AC $\neq$ 0),
SPA Skip on Positive Accumulator (skip the next instruction if AC $\geq$ 0),
SNA.SPA Skip on Non–Zero and Positive Accumulator (skip the next instruction if AC > 0).

The EDUC–8 OPR (operate) instructions are of the form 7XY, where 7 indicates OPR, X is a two bit value corresponding to MB[4:3] and Y is a three bit value corresponding to MB[2:0]. We note that the combined instructions typically use MB[3:2] = 3, e.g., for CLA.CMA and SZA.SMA. Similarly, we use the same value for STL = CLL.CML, SKP = SZL.SNL and SNA.SPA. Bit MB[4] is used to select a group of instructions. In order to add the additional instructions we combine MB[4] with MB[1:0] to form eight separate groups. The table below shows the eight groups, along with either three or four instructions in each group. The new instructions are in red.

Instruction Group MB4 MB1 MB0	MB3 MB2			
	00	10	01	11
000	NOP	CLA	CMA	CLA.CMA
001	IAC	CLA.IAC	CMA.IAC	DAC
010	RAL	OSR	BSW	–
011	RLL	CLL	CML	STL
100	–	SZA	SMA	SZA.SMA
101	HLT	RTL	RTR	–
110	RAR	SNA	SPA	SNA.SPA
111	RLR	SZL	SNL	SKP

Note that DAC is not really a combined instruction, as CLA.CMA.IAC is effectively the same as CLA. Instruction 720 (group 4 instruction 0) is effectively a NOP (no operation) instruction, but since we already have 700 for NOP, we leave this instruction free for future use.

E8L/T Timing Board Modifications

For a TAD or IAC instruction, the EDUC–8 uses a 9001 (equivalent to the 74105) register that is implement as a D–type flip flop (DFF). This DFF will also be used for DAC and is used to store the carry out from a one bit adder (a 7480), using the output of the DFF as the carry in to the adder. Since the adder and DFF are also used for other operations, such as incrementing the program counter (PC) register, this DFF can not be used for the link bit L. Conveniently, we can replace the 9001 with a 7474 dual DFF where one DFF (called CI for carry in) replaces the function of the 9001 and the other DFF is used for L.

The timing of the TAD, IAC and DAC instructions both occur during T2 to T9 of the 24 clock pulses during an EXECUTE operation. Thus, after T9 we need to invert L if CI is high. The next available signal to do this is T13. However, signal $\overline{T0+T12}$ is used to asynchronously clear CI so that the adder can be used during T14 to T21 for any skip or jump instruction! Thus, we need to create a new signal before T12 and after T9 to perform the update.

The signal that we chose is T10. The output of the 74154 one of 16 decoder is $\overline{S0}$ to $\overline{S11}$ and the 7473 JK–type FF Sequence Counter is T0–T11 and T12–T23. The operation we require is T10 = S10.T0–T11. Signal S10 is already available, as it used for T22·5.

We note that signal T14–21 is duplicated on E8/T. We have that T14–21(10) goes to E8/D(9) and E8/M(8) and T14–21(12) to E8/D(10), where the number in brackets is the edge connector pin

number. This replication of signals is a legacy from the first version of EDUC–8 which only had 32 bytes of RAM, where T17–21(12) was originally used. Since T14–21(10) goes to both E8/D and E8/M we can disconnect T14–21(12) and use it as a spare pin. Also, we reuse the 7400 2–input NAND (NAND2) and 7400 inverter (INV) to generate T10 from S10 (replacing S2–9) and T0–11 (replacing T12–23). To reduce E8L/C line lengths, we connect T10 to spare pin (32). Note that (32) is connected to GND on E8/C, which will need to be disconnected.

Another reason for this change is that E8/D only has one spare pin available, pin 32. By disconnecting T14–21 from E8/D(10) and connecting it to E8/D(9), this frees up E8/D(10), allowing two spare pins to be created. The required changes are shown in schematic E8L/T & E8L/D. The chip and pin numbers follow that used by [2].

For RAL and RAR instructions, this is performed during T1 and T13, respectively. The same timing is also used for RLL and RLR. For RTL and RTR, two rotations are performed during T1 and T13. Thus, we output signal S1 = T1+T13 to spare pin (22) from an existing 7404 INV gate, as shown in schematic E8L/T & E8L/D.

For the BSW instruction, this is equivalent to rotating AC four times. The signal we chose to do this is Q2 from the 74161 counter. Signal Q2 is equivalent to T4–7+T20–23. This signal is later ANDed with T2–9 to generate the required rotate signal. Q2 is connected to spare pin (12), going to E8/D(10) using the existing line on E8/C previously used for T14–21.

E8L/D Decoder Board Modifications

Here we need to modify the decoder board to generate the required control signals for the 16 new instructions. The original instruction format is below.

MB7	MB6	MB5	MB4	MB3	MB2	MB1	MB0
1	1	1	0	CLA	CMA	RAL	IAC
1	1	1	1	SZA	SMA	RAR	HLT

As RAL and RAR are similar to RLL and RLR, CLA and CMA are similar to CLL and CML and SZA and SMA are similar SZL and SNL, we would like to reuse these bit positions if possible. We can do this by noting that RAL and RAR are not used in any combined instruction. Thus, in order to select the link instructions we have MB1 (normally used to indicate a rotate) and MB0 (normally used for IAC and HLT) are both 1. The instruction format for the eight link instructions are as follows, where bits MB5 to MB7 are not shown, since they are alway 1. We then added the other new eight instructions to the existing and link instructions.

	MB4	MB3	MB2	MB1	MB0
RLL	0	0	0	1	1
	0	CLL	CML	1	1
RLR	1	0	0	1	1
	1	SZL	SNL	1	1

The E8/D board produces the following eight signals, corresponding to the original eight basic instructions, where LO = EXECUTE.Q7./MB4, HI = EXECUTE.Q7.MB4 and Q7 is the output from the 9334/74259 1 of 8 decoder latch corresponding to MB[7:5] = 7 during FETCH.

/CLA = $\overline{\text{LO.MB3}}$
/CMA = $\overline{\text{LO.MB2}}$
/RAL = $\overline{\text{LO.MB1}}$
/IAC = $\overline{\text{LO.MB0}}$
/SZA = $\overline{\text{HI.MB3}}$
/SMA = $\overline{\text{HI.MB2}}$

/RAR = $\overline{\text{HI.MB1}}$
/HLT = $\overline{\text{HI.MB0}}$

These eight signals need to be modified and a further seven control signals added using the truth tables below. The first column of the truth tables gives the signal name, followed by the instructions that use this signal. The second column gives the lower five bits of the instruction, equal to MB[4:0] during EXECUTE.

For group MB4,MB1,MB0 = 0 or 1 instructions the four signals used are as follows. For DAC, we use /CLA and /CMA to place value –1 onto B–BUS. However, we need to disable /IAC and send AC onto C–BUS. B–BUS and C–BUS are added together in E8/P and sent to A–BUS where it is shifted into AC.

/CLA	MB[4:0]	/CMA	MB[4:0]	/IAC	MB[4:0]	/DAC	MB[4:0]
CLA	01000	CMA	00100	IAC	00001	DAC	01101
CLA.CMA	01100	CLA.CMA	01100	CLA.IAC	01001		
CLA.IAC	01001	CMA.IAC	00101	CMA.IAC	00101		
DAC	01101	DAC	01101				

We thus have

/CLA = $\overline{\text{LO.MB3./MB1}}$
/CMA = $\overline{\text{LO.MB2./MB1}}$
/IAC = $\overline{\text{LO./(MB3.MB2)./MB1.MB0}}$ = $\overline{\text{LO./H3.L1}}$
/DAC = $\overline{\text{LO.H3.L1}}$

where H3 = MB3.MB2 and L1 = /MB1.MB0. Signals /CLA, /CMA and /IAC are all used to shift AC. As /CLA and /CMA are also used for DAC, we have the required shift operation for DAC, as well as placing –1 onto B–BUS. Instructions OSR and BSW also require AC to be shifted. For BSW the shift needs to be four, instead of eight clock cycles. We thus modify /CLA to be /(CLA+OSR+BSW./Q2). We use /Q2 instead of Q2 to save an INV. As /(CLA+OSR+BSW./Q2) is ANDed with T2–9, this implies four shifts occur from T2–3 and T8–9.

The group 2 signals are as follows. To save on logic, the /RAL signal is replaced with RAL which is also used for RLL. Signal MB0 is later used to select whether to shift with the link bit. Note that OSR.BSW is not a valid instruction.

RAL	MB[4:0]	OSR	MB[4:0]	BSW	MB[4:0]
RAL	00010	OSR	01010	BSW	00110
RLL	00011	OSR.BSW	01110	OSR.BSW	01110

We thus have

RAL = LO./MB3./MB2.MB1 = LO.H0.MB1
OSR = LO.MB3.MB1./MB0 = LO.MB3.L2
BSW = LO.MB2.MB1./MB0 = LO.MB2.L2

where H0 = /MB3./MB2 and L2 = MB1./MB0. The signals that are output are RAL, OSR (used to shift SR onto A–BUS) and /(OSR+BSW./Q2) (used to shift AC onto A–BUS).

The group 3 signals are

/CLL	MB[4:0]	/CML	MB[4:0]
CLL	01011	CML	00111
CLL.CML	01111	CLL.CML	01111

We thus have

/CLL = $\overline{\text{LO.MB3.MB1.MB0}}$ = $\overline{\text{LO.MB3.L3}}$
/CML = $\overline{\text{LO.MB2.MB1.MB0}}$ = $\overline{\text{LO.MB2.L3}}$

where L3 = MB1.MB0. The operation for /CLL is that it is used to asynchronously clear L during T0·5 followed later by /CML during T10. To reduce logic we output /(CLL.T0·5), where T0·5 is available from E8/D.

All the skip instructions use the /SZA and /SMA signals, where MB1 and MB0 are later used to select the appropriate signals to perform the skip. We have for groups 4, 6 and 7 the truth table below. As for RAL, we use RAR instead of /RAR to reduce logic. Similarly, MB0 is used to later select between RAR and RLR.

/SZA	MB[4:0]	/SMA	MB[4:0]	RAR	MB[4:0]
SZA	11000	SMA	10100	RAR	10010
SZA.SMA	11100	SZA.SMA	11100	RLR	10011
SNA	11010	SPA	10110		
SNA.SPA	11110	SNA.SPA	11110		
SZL	11011	SNL	10111		
SZL.SNL	11111	SZL.SNL	11111		

We thus have

/SZA = $\overline{\text{HI.MB3./(/MB1.MB0)}}$ = $\overline{\text{HI.MB3./L1}}$
/SMA = $\overline{\text{HI.MB2./(/MB1.MB0)}}$ = $\overline{\text{HI.MB2./L1}}$
RAR = HI./MB3./MB2.MB1 = HI.H0.MB1

Lastly, we have the truth table for the group 5 signals. Note that RTL.RTR is not a valid instruction. As MB0 = 1 for RTL and RTR, this implies that the rotation is through the link bit, as desired. The RTL and RTR signals are used to select the T1+T13 signal.

RTL	MB[4:0]	RTR	MB[4:0]	/HLT	MB[4:0]
RTL	11001	RTR	10101	HLT	10001
RTL.RTR	11101	RTL.RTR	11101		

We thus have

RTL = HI.MB3./MB1.MB0 = HI.MB3.L1
RTR = HI.MB2./MB1.MB0 = HI.MB2.L1
/HLT = $\overline{\text{HI./MB3./MB2./MB1.MB0}}$ = $\overline{\text{HI.H0.L1}}$

In order to generate the modified and new signals, we need to replace the two 7400 quad NAND2 chips that are used for /CLA, /CMA, /RAL and /IAC (U10) and /SZA, /SMA, /RAR and /HLT (U11) with two 7410 triple NAND3 chips. We also need to add one 7400, one 7402 quad NOR2, one 7420 dual NAND4 and two 7427 triple NOR3 chips.

As explained previously, E8/D only has one spare edge connector pin E8/D(32), which we connect to /(CLL.T0·5). As T14–21 from pin 9 and T14–21 from pin 10 are logically the same, we can disconnect T14–21 from pin 10 and connect it to pin 9. The frees up pin 10 so that it can be connected to Q2 from E8L/T. Figure E8L/T & E8L/D shows the modification.

Also shown in Figure E8L/T & E8L/D are the inverted signals of MB0, MB1, MB2 and MB3. We note that /MB3 is the same as IO_DEVICE_SELECT connected to pin (4). The 7404 hex INV U14 has three spare gates, which we use for /MB0, /MB1 and /MB2.

Figure E8L/D shows the logic required to generate the 15 control signals. The original eight control signals use their same pin connections, with /RAL and /RAR being renamed to RAL and RAR. For positive logic signals, NOR gates are generally used, which are equivalent to AND gates with inverted inputs. This means that /LO and /HI signals were needed, which fortunately were already available.

For U10 and U11 we tried to arrange the signals to minimise the number of new connections, even though they are using different chips (7410 instead of 7400). For U10, /IAC and /CMA are in the same pin locations. For U11, /SMA is in the same pin location. One spare 7410 NAND3 gate U2A is used for /(CLL.T0·5).

When the design was completed, it was found that MB1 had 12 UL, exceeding the 10UL output of the 7495 register used for MB[7:0]. There were five loads on E8L/D, two loads on E8/M, four loads on E8L/P and one load on E8/F. This meant we had to add another INV and would have required adding an additional chip, as there were no spare gates available. However, an INV was being used to invert /T0·5 from pin (8) to generate T0·5 on E8/D. As EDUC–8ME had modified E8/M to output T0·5, we decided to connect this T0·5 to pin (8), giving us the required INV, located at 7404 U14F. This INV inverts /MB1 to MB1A, reducing the total load on MB1 to 9 UL.

E8L/A Accumulator Board Modifications

For a RAR instruction, AC0 is placed on B–BUS, which goes to the A1 input of the serial 7480 adder on E8/P. During T13, the AC input of the adder is enabled so that the summation output (equal to AC0) goes to the A–BUS and then to the DS serial input of the left 7495 shift register U1 on E8/A. The clock input $\overline{\text{CP1}}$ of the shift register is also enabled during T13, so that a single shift of AC to the right then occurs on the falling edge of MCPB. For a RLR instruction, the RAR signal also goes high. As the link bit L is located on E8/P, for a RLR we simply disable the output of the adder and instead place L onto the A–BUS. Thus no modifications are required on E8/A for RLR.

The signal /(T13.RAR) from E8/P is used to control $\overline{\text{CP1}}$ for the shift right. However, we need to change this to /(T13.RAR+(T1+T13).RTR) so that the RTR operation can be performed during T1 and T13. A 2–2 AND–OR–INV (AOI) gate (half of a 7451) is used to perform this on E8L/A. RAR is connected to pin (D) (previously called /RAR), T13 is connected to spare pin (16) which needs to be disconnected from GND, T1+T13 is connected to spare pin (3) which needs to be disconnected from VCC and RTR is connected using an external lead.

To control AC0 going onto B–BUS, the signal RAR+TAD = $\overline{\text{/RAR./TAD}}$ is used. Here we simply replace /RAR with /(T13.RAR+(T1+T13).RTR).

For a RAL instruction, signal RAL is connected to the parallel enable (PE) input of the shift register with $\overline{\text{CP2}}$ connected to T1, with the parallel inputs rotating AC to the left on the falling edge of T1 when PE is high. Let P[7:0] be the parallel input of the two 7495 4–bit registers for AC. For a RLL instruction, we need to send AC7 to the link bit FF (L) on E8/P and the L output from E8/P to P0 on E8/A. This can be achieved by disconnecting AC7 from P0 and placing AC7 onto the B–BUS (similar to RAR for AC0) when RAL is high using a spare 7401 OC NAND2 U5C. That is

$$\text{B–BUS} = \overline{\text{AC7.RAL}}$$

For a RAL instruction, AC7 on B–BUS goes through the adder on E8/P and onto the A–BUS. By connecting /A–BUS to P0 we have the signals in their correct position. For a RLL instruction,

we instead place L onto the A–BUS. However, we can't use T1 directly as L could change on the falling edge of MCPB (when it gets updated with AC7 during a RLL instruction) before the falling edge of T1. To fix this problem, we need to gate MCPB with T1 to $\overline{CP2}$ (similar to what is done with $\overline{CP1}$). That is

$\overline{CP2}$ = MCPB.T1

To implement $\overline{CP2}$ we need another 7400 NAND2 and 7404 INV. A spare INV is available from U6C (its input is connected to the original RAL signal, which needs to be disconnected). To obtain the NAND2 we observe that the inverter for $\overline{AND+DCA+IOT\_SHIFT}$ is implemented using 7400 gate U7B. By replacing this gate with a spare 7405 OC INV U4B we thus obtain the required gate.

Similar to RAR, we also need to shift left for RTL during T1 and T13. This can be done using the other half of the 7451 AOI where we replace RAL with T1.RAL+(T1+T13).RTL. That is, we now have

PE = T1.RAL+(T1+T13).RTL
B–BUS = $\overline{AC7.(T1.RAL+(T1+T13).RTL)}$
$\overline{CP2}$ = MCPB.(T1.RAL+(T1+T13).RTL)

The 7404 INV U6E that was used for inverting /RAL to RAL is now used to invert /(T1.RAL+(T1+T13).RTL) from the 7451 AOI. Thus, pin (F) needs to be disconnected from U6E. Signals RAL (F), T1 (13) and T1+T13 (3) are already available. Signal RTL is connected using an external lead from E8L/D.

We also generate signal ROTATE as follows using a NAND2.

ROTATE = $\overline{/(T1.RAL+(T1+T13).RTL)./(T13.RAR+(T1+T13).RTR)}$
= T1.RAL+T13.RAR+(T1+T13).(RTL+RTR)

The ROTATE signal is used by E8L/P in implementing the various shift operations. To obtain the 7400 NAND2 we use U7C that was used to invert CP1 to /CP1 for the 7495 AC shift register. To replace U7C for /CP1, we use 7404 INV U6B that was used for inverting /DCA to DCA, which is obtained using the modification below. ROTATE is connected to pin (8), previously used for /(T13.RAR), where the signal goes to E8L/P instead of the other way around.

For a DAC instruction we have signals CLA = 1, CMA = 1, IAC = 0 and DAC = 1 (using positive logic). The first two signals places all ones (effectively –1) onto B–BUS. Note that our convention is to use positive logic when discussing values on the various buses. The actual values will be inverted. We also need to place AC0 onto C–BUS, so that it can be serially added with B–BUS in E8L/P.

Here we use the 7401 OC NAND2 U10C that was used to invert /(/CLA./CMA.IAC) to /CLA./CMA.IAC and replace it with spare 7405 OC INV U4C. U10C is used to replace 7404 INV U6B (to be used for inverting CP1 to /CP1 above) which is used to place AC0 onto C–BUS. The original circuit was

C–BUS = $\overline{AC0.DCA}$ (7401 OC NAND2 U10D)
DCA = $\overline{/DCA}$ (7404 INV U6B)

which now becomes

C–BUS = $\overline{AC0.(DCA+DAC)}$ (7401 OC NAND2 U10D)
DCA+DAC = $\overline{/DCA./DAC}$ (7401 OC NAND2 U10C)

Thus, when DAC goes high, AC0 is placed onto C–BUS as desired. Signal /DAC is externally connected from E8L/D.

The last modification for E8L/A is to place AC0 onto A–BUS when /(OSR+BSW./Q2) is low. As /CLA has been replaced with /(CLA+OSR+BSW./Q2) using existing pin (H) this performs the required shifting of AC (eight times for OSR and four times for BSW). The existing circuit for A–BUS is

A–BUS = $\overline{\text{AND.(D–BUS.AC0)}}$ (7401 OC NAND2 U5B)
AND = $\overline{\text{/AND}}$ (7404 INV U6A)
D–BUS.AC0 = $\overline{\text{/(D–BUS.AC0)}}$ (7400 NAND2 U9B)

By swapping the inverters for AND and D–BUS.AC0, the circuit now becomes

A–BUS = $\overline{\text{(AND+OSR+BSW./Q2).(D–BUS.AC0)}}$ (7401 OC NAND2 U5B)
AND+OSR+BSW./Q2 = $\overline{\text{/AND./(OSR+BSW./Q2)}}$ (7400 NAND2 U9B)
D–BUS.AC0 = $\overline{\text{/(D–BUS.AC0)}}$ (7404 INV U6A)

We note that during EXECUTE for an OSR or BSW instruction, that D–BUS is pulled up to a high value since the $\overline{\text{CS}}$ inputs to the memories will be high. In EDUC–8, control signal MEMORY_ENABLE is used to gate the $\overline{\text{CS}}$ inputs, which are only enabled when data is read or written to the memory. In EDUC–8ME, MEMORY_ENABLE is used to gate DOUT from the external memory onto D–BUS. Thus, if OSR+BSW./Q2 is high, AC0 is placed onto the A–BUS. /(OSR+BSW./Q2) is externally connected from E8L/D.

In summary, the modifications require four external connections and only one additional 7451 AOI chip. Figure E8L/A gives the schematic for the required changes and additions to E8/A.

E8L/P Program Counter and Adder Board Modifications

To implement the link bit we note that the 9001/74105 JK FF used for the serial adder carry is implemented as a D–type FF. By replacing this FF with a 7474 dual D–FF (which also has independent asynchronous set and reset inputs like the 9001) we implement both the carry FF and L in one chip.

To implement the SZL and SNL instructions, we note that signals SZA and SMA are ANDed with signals H⊃AC=0 and H⊃AC=–VE, respectively. These signals are equal to $\overline{\text{AC0+AC1+AC2+AC3+AC4+AC5+AC6+AC7}}$ and AC7, respectively. To simplify the notation we call these signals ACZ and AC7, respectively. Gates 7400 NAND2 U11B and U11A provide the outputs /(SZA.ACZ) and /(SMA.AC7), respectively. We disconnect ACZ and AC7 from these gates and connect the gate inputs to two of the outputs (ZA and ZB) of a 74157 multiplexer, where the new signal MB0 is connected to the select input of the 74157. Gates U11B and U11A are now implemented as /(SZA.ZA) and /(SMA.ZB), respectively.

Ignoring for the moment the SNZ and SPA instructions, for the multiplexer outputs we have

ZA = /MB0.ACZ + MB0./L
ZB = /MB0.AC7 + MB0.L

Thus, when MB0 is low, the carry FF will be set during T13 for a normal SZA (ACZ high) or SMA (AC7 high) instruction. For MB0 high, the carry FF will be set during T13 for SZL (/L high) or SNL (L high) instruction. During T14–21, the program counter PC is passed through the serial adder and will be incremented if the carry FF is high, causing a jump over the next instruction.

For a SNZ or SPA instruction (where MB0 is also 0, but MB1 is 1), we select either /ACZ or /AC7 for ZA and ZB. However, for a SNZ.SPA instruction the operation is skip on non–zero AC

and positive AC. That is, skip if /ACZ./AC7 is high and not /ACZ+/AC7. The latter logic would be equivalent to a SKP instruction. We modify ZA and ZB so that

ZA = /MB0.I0A + MB0./L
ZB = /MB0.I1A + MB0.L

where we have the logic table

MB1	SZA	SMA	I0A	I0B
0	0	0	–	–
0	1	0	ACZ	–
0	0	1	–	AC7
0	1	1	ACZ	AC7
1	0	0	–	–
1	1	0	/ACZ	–
1	0	1	–	/AC7
1	1	1	0	/ACZ./AC7

A solution for the above table is

I0A = /(MB1.SMA).(ACZ⊕MB1)
I0B = /(MB1.SZA.ACZ).(AC7⊕MB1)

We implement the above logic using a 7422 dual OC NAND4 and half of a 74136 quad OC XOR2. Here we use the OC outputs to perform the "AND" operation. Both I0A and I0B have two standard outputs and one standard input. Using the pullup equations given in [3], for 74 series TTL pullup values range from 337 to 4352 Ω. For 74LS pullup values range from 625 to 9318 Ω. We thus use a value of 2.2K for the pullups, like that used for other OC gates with a single outputs.

In the original design, signal /(T13.RAR) is used to enable the AC input of the serial adder (connected to B–BUS) and the serial adder output to A–BUS. For a RAR instruction, this effectively passes AC0 from B–BUS, through the adder and onto A–BUS during T13.

For the modified design we instead must enable the adder for RAL and RAR instructions during T1 or T13 and disable the adder for RLL, RLR, RTL and RTR instructions. We connect signal ROTATE to pin (9) (previously used to output /(T13.RAR)) which is output from E8L/A. Spare gate 7404 INV U12E is used to invert ROTATE to /ROTATE. Note that signal /(T14–21.(JMP+JMS)) needs to be disconnected from the input of U12E.

Here we have the 74157 multiplexer output ZC as

ZC = /MB0./ROTATE + MB0.HI
= MB0 + /ROTATE

where HI is connected to +5V via pullup resistor R2. Thus, when MB0 is low a normal RAL or RAR instruction is performed. When MB0 is high for a RLR, RLL, RTL or RTR instruction, the output from the adder is disabled. Signal /(T13.RAR) normally goes to one input of each of the two 7420 4–input NAND gates (U13A and U13B). To make the modification, /(T13.RAR) needs to be disconnected from the gates and ZC connected.

We also want to place L onto A–BUS for RLR, RLL, RTL or RTR instructions. Here we have the 74157 multiplexer output ZD as

RLI = /MB0.GND + MB0.ROTATE
= MB0.ROTATE

where A–BUS = /(RLI.L) (implemented using a 7401 OC NAND2) and RLI is the rotate link signal. Thus when MB0 is low, the A–BUS output is disabled for a RAR and RAL instruction. For MB0 high, we output L onto the A–BUS during T1 for a RLL instruction, during T13 for a RLR instruction and during T1 and T13 for RTL or RTR instructions.

For the clock input to L, we gate this with MCPB, similar to that for the carry FF. We have the clock input LCP = /(LE.MCPB) where the link enable signal LE when high enables the clock. LE goes high if RLI is high or during T10 if CML is high or if CARRY (the serial adder FF output) is high for a TAD, IAC or DAC instruction. That is

LE = RLI+T10.IL
IL = CML+CARRY.(TAD+IAC+DAC)

where IL is the invert link signal. LCP is generated using the 7400 NAND2 U10A that was previously used for /(T13.RAR). Signal LE is generated using a 7451 2–2 AOI and 7404 INV U9D that was previously used for RAR. IL is generated using two 7400 NAND2 gates U7C and U7D and 7410 NAND3 U3A where TAD+IAC+DAC = $\overline{/TAD./IAC./DAC}$.

U7C is a spare gate. U7D was previously used to invert A–BUS to /A–BUS for input to the DS input of the 7496 register used for PC[7:5]. A 74136 OC XOR2 is used instead for /A–BUS. U3A was previously used for signal T2–9.(JMS+F+D+E) where only two inputs are used. To generate T2–9.(JMS+F+D+E), we instead use a 7401 OC NAND2.

For the D input for L we use a 7451 2–2 AOI where

LD = $\overline{B–BUS.RLI+IL.L}$

Thus, if RLI is high we are performing a link shift operation and place the data on B–BUS (equal to AC0 or AC7) into L. B–BUS is inverted as desired. If IL is high, this implies that we are inverting L, as desired.

The last signal for the L FF is /(CLL.T0·5), which asynchronously clears L during T0·5 if CLL is high. As CML follows CLL, a CLL.CML = STL instruction will set L. A 74136 OC XOR2 is used to output /LINK to the front panel.

For a DAC instruction, we need to enable the input of B–BUS and C–BUS of the 7480 adder which is connected to input A1 and B2, respectively. The enable inputs of the adder are AC and BC, respectively. As CMA is high for DAC, this enables AC of the adder during T2–9. The adder output is also enabled so that it is output to A–BUS. As the adder is not enabled when /(CLA+OSR+BSW./Q2) is low, this leaves A–BUS free.

One of the control inputs for BC is /(T2–9.TAD) using 7400 NAND2 U6D. We replace the 7404 INV U9C used to invert /TAD to TAD with a 7401 OC NAND2 which outputs TAD+DAC = $\overline{/TAD./DAC}$ and connect this to U6D which now outputs /(T2–9.(TAD+DAC)). Thus, when DAC goes this will enable BC so that C–BUS (AC) is added to B–BUS (–1).

To send the switch register SR[7:0] onto A–BUS for an OSR instruction we use a 74166 8–bit parallel to serial converter. We could not use the existing 7495 parallel to serial converters on E8/M to do this, as these are the registers used for MB[7:0], which holds the instruction. The SR[7:0] inputs are connected to the parallel inputs of the 74166. Signal T2–9 is connected to the active low parallel enable $\overline{PE}$ input. Thus, when T2–9 is low, SR[7:0] is input. When T2–9 goes high, the data is shifted out least significant bit first. As the 74166 clocks on the rising edge, we need to invert MCPB. This is done using 7404 INV U9C, which was previously used for signal TAD.

A 7401 OC NAND2 is used to output the serial output register Q7 of the 74166 onto A–BUS when OSR is high, i.e., A–BUS = /(Q7.OSR). As AC is also output to A–BUS, we have the required

"OR" operation of AC+SR, which gets shifted into AC. Since the inverted logic OC outputs are "ANDed" together and then inverted before input to AC, the logical operation is $\overline{/AC./SR}$ = AC+SR.

The above modifications require one each of 7401 quad OC NAND2, 7422 dual OC NAND4, 7451 dual 2–2 AOI, 74136 quad OC XOR2, 74157 quad MUX2 and 74166 8–bit parallel to serial converter, for a total of six additional chips. Signals MB0, MB1 and /LINK are connected to spare pins (P), (Q) and (R), respectively. For pins (Q) and (R), these need to be disconnected from GND.

Signals T10 and /(CLL.T0·5) are connected to spare pins (29) and (32), respectively. Signals /DAC, /CML and OSR are connected to E8L/D using external lines. Figure E8L/P gives the schematic for the required changes and additions to E8/P.

E8L/F Front Panel and E8L/C Connector Board

The front panel needs one small modification to add a light emitting diode (LED) and pullup resistor for the link bit. This LED should be placed to the left of the LED used for AC7.

Where spare pins are used on the board, a number of changes are required for the connector board. These changes are shown in Figure E8L/C & E8L/F. We also include the additional 32–pin and 16–pin edge connectors for EDUC–8ME. Three 3–pin jumpers are also included, so that EDUC–8L can be used with or without EDUC–8ME.

A–BUS has two additional outputs added on E8L/P and one output for E8ME/B. B–BUS has one additional output added on E8L/A and one input added on E8L/P. Thus, we need to check the values of the pull–up resistors. This is summarised in the table below. We also give pullup values for C–BUS and two sets of values for D–BUS, one set with two 93415 1Kx1 memories as used in EDUC–8 and one set for EDUC–8ME, where only a single 7401 OC gate is used. The 93415 has I_{OH} = 0.1 mA (leakage current), I_{OL} = 16 mA and V_{OL} = 0.45 V.

For the unit loads (UL), the first value corresponds to the number of 0.04 mA input high unit loads. The second value corresponds to the number of 1.6 mA input low unit loads. The average R_{av} is over the largest R_{min} and smallest R_{max} for 74 and 74LS. Note that the 7480 operand inputs (which does not have a 74LS equivalent) has an unusual 0.4 UL high. To reduce propagation delay and to simplify ordering, we choose 1K for all the pullup values. Note that the original pullup values are too low for 74LS and may cause improper operation.

Table 1. E8L/C R1 to R4 Pullups

		74 series TTL			74LS series TTL			74/74LS	
	N_O	*UL*	R_{min} (Ω)	R_{max} (Ω)	*UL*	R_{min} (Ω)	R_{max} (Ω)	R_{av} (Ω)	R_{rec}
R1 (A–BUS)	6	2/2	379	1487	1/0.5	660	3203	1074	1K
R2 (B–BUS)	6	1.4/2	379	1510	0.9/1.25	792	3223	1151	1K
R3 (C–BUS)	6	2.4/3	433	1472	1.4/1.5	848	3125	1160	1K
R4 (D–BUS)	2	3/3	429	7344	1.5/0.75	324	7885	3887	1K
R4 (D–BUS)	1	3/3	433	6351	1.5/0.75	699	12812	3525	1K

For the switch register SR[7:0], this now has one additional load, increasing the total UL to four. There is one load on E8/M used for loading MB[7:0], two loads on E8/P for loading PC[7:0] and the OSR shift register and one load used for EDUC–8ME. The maximum pullup value is then 58.75/4 = 14.688 kΩ [3], which implies that the existing 10K pullups can continue to be used.

Signal Loading

The table below gives the signal loading for all new and modified signals. As most TTL outputs have 20 UL high and 10 UL low, with most inputs having 1 UL high and 1 UL low, the number of standard inputs that can be driven is 10. We see that all signals meet this requirement. If a signal is connected to a pin, this is given as the first number, followed by a dash and then the number of ULs. Pin X indicates an external connection. The page zero modification [5] is included in E8/M for signals MB3 and MB4.

Signal Name	E8/T	E8/D	E8/M	E8/P	E8/A	E8/IOT	E8/F	E8ME/B	Total
EXECUTE	23–0	21–10							10
EXEC	26–1							22–2	3
MCPB	11–1			11–4	9–2	10–1		10–2	10
$\overline{\text{MR}}$	25–1							21–2	3
RST	2								2
$\overline{\text{RST}}$	10								10
Q2	12–1	10–1							2
S10	2								2
T0·5		8–2	21–2					19–1	5
$\overline{\text{T0·5}}$	8–0		7–3						3
T0–11	6								6
T1	16–0		12–1	14–1	13–1	15–1		17–1	5
T1+T13	22–2				3–2				4
T2–9	15–0	12–3	11–1	13–3	12–1	14–1		16–1	10
T10	32–0			29–1					1
T13	14–0		10–1	12–2	16–1	13–1		15–1	6
T14–21	10–0	9–3	8–1						4
HI		3							3
$\overline{\text{HI}}$		4							4
L1		3							3
$\overline{\text{L1}}$		5							5
LO		4							4
$\overline{\text{LO}}$		4							4
MB0		J–3	J–3	P–1			X–1		8
MB1		K–2	K–2	Q–4			X–1		9
MB1A		3							3
MB2		L–7	L–2				X–1		10
MB3		M–5	M–3				X–1	M–1	10
MB4	31–1	N–2	N–4				X–1	N–2	10
$\overline{\text{MB0}}$	2								2
$\overline{\text{MB1}}$	4								4
$\overline{\text{MB2}}$	4								4
$\overline{\text{MB3}}$		4–5				5–3		5–1	9
$\overline{\text{MB4}}$		5–1				6–3		6–1	5
$\overline{\text{CLA+OSR+BSW./Q2}}$		31–0			H–3				3
$\overline{\text{CLL.T0·5}}$		32–0		32–2					2
$\overline{\text{CML}}$		X–0		X–1					1

Signal Name	E8/T	E8/D	E8/M	E8/P	E8/A	E8/IOT	E8/F	E8ME/B	Total
$\overline{\text{DAC}}$		X–0		X–2	X–1				3
$\overline{\text{IAC}}$		28–1		F–2	E–1				4
OSR		X–2		X–1					3
$\overline{\text{OSR+BSW./Q2}}$		X–0			X–1				1
RAL		29–0			F–1				1
RAR		25–0			D–1				1
ROTATE			9–2	8–0					2
RTL		X–0			X–1				1
RTR		X–0			X–1				1
$\overline{\text{TAD}}$		B–2		B–2	B–1		X–1		6
AC0 (AC_BIT_0)					11/J–7	12–1	X–1	14–1	10
AC7 (H⊃AC=–VE)				H–1	R–2		X–1		3
H⊃AC=0				16–2	15–0				2
$\overline{\text{LINK}}$				R–0			X–1		1

Program Encoding Guide

Like for the original EDUC–8, a programming guide is given in the following page. This includes the existing instruction set, along with the new instructions included with EDUC–8ME and EDUC–8L.

References

[1] S. S. Pietrobon, “EDUC–8ME: Memory and input/output port extension for the EDUC–8 microcomputer,” 8 Sep. 2024. http://www.sworld.com.au/steven/educ–8/educ–8me.pdf

[2] G. Sutter, “EDUC–8 schematics,” 6 June 2024. https://www.mediafire.com/file/ouvtn5e342jknjl/educ8_gs_schematics.zip/file

[3] Fairchild, “TTL Data Book,” 1978.

[4] S. S. Pietrobon, “Using low power Schottky (74LS) logic for the EDUC–8 microcomputer,” 29 Mar. 2023. http://www.sworld.com.au/steven/educ–8/educ8–74ls.pdf

[5] S. S. Pietrobon, “Page Zero Implementation for EDUC–8,” 2 Mar. 2023. http://www.sworld.com.au/steven/educ–8/educ8_page_zero.pdf

EDUC–8L PROGRAM ENCODING GUIDE

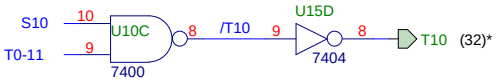
Code	Mnemonic	Operation
		MEMORY REFERENCE INSTRUCTIONS
0XY	AND	Logical AND
1XY	TAD	Two's complement add
2XY	ISZ	Increment and skip if zero
3XY	DCA	Deposit and clear AC
4XY	JMS	Jump to subroutine
5XY	JMP	Jump
	X=0	Direct to Y (page zero for AND, TAD, ISZ, DCA)
	X=1	Direct to 8+Y
	X=2	Indirect to Y (page zero for AND, TAD, ISZ, DCA, JMS)
	X=3	Indirect to 8+Y
		INPUT/OUTPUT TRANSFER (IOT) INSTRUCTIONS
601	SKF 0	Skip on input ID0 flag
602	KRS 0	Read input ID0 buffer
604	RKF 0	Reset input ID0 flag
606	KRB 0	Read input ID0 buffer, reset flag
611	SKF 1	Skip on input ID1 flag
612	KRS 1	Read input ID1 buffer
614	RKF 1	Reset input ID1 flag
616	KRB 1	Read input ID1 buffer, reset flag
621	SDF 0	Skip on output OD0 flag
622	LDS 0	Load output OD0 buffer
624	RDF 0	Reset output OD0 flag
626	LDB 0	Load output OD0 buffer, reset flag
631	SDF 1	Skip on output OD1 flag
632	LDS 1	Load output OD1 buffer
634	RDF 1	Reset output OD1 flag
636	LDB 1	Load output OD1 buffer, reset flag
		EDUC–8ME INSTRUCTIONS
603	IFR	Instruction field read
607	IFL	Instruction field load
613	IDR	Input device field read
617	IDL	Input device field load
623	DFR	Data field read
627	DFL	Data field load
633	ODR	Output device field read
637	ODL	Output device field load

Code	Mnemonic	Operation
		OPERATE (OPR) MICROINSTRUCTIONS
700	NOP	No operation
701	IAC	Increment AC
702	RAL	Rotate AC one bit left
703	RLL	Rotate L and AC one bit left
704	CMA	Complement AC
705	CMA.IAC	Complement and increment AC (two's complement)
706	BSW	Byte swap (exchange AC[7:4] with AC[3:0])
707	CML	Complement L
710	CLA	Clear AC
711	CLA.IAC	Set AC to contain 1
712	OSR	OR switch register with AC
713	CLL	Clear L
714	CLA.CMA	Set AC to contain –1
715	DAC	Decrement AC
717	STL	Set L
721	HLT	Halt at end of execute cycle
722	RAR	Rotate AC one bit right
723	RLR	Rotate L and AC one bit right
724	SMA	Skip on minus AC
725	RTR	Rotate L and AC two bits right
726	SPA	Skip on positive AC
727	SNL	Skip on non–zero L
730	SZA	Skip on zero AC
731	RTL	Rotate L and AC two bits left
732	SNA	Skip on non–zero AC
733	SZL	Skip on zero L
734	SZA.SMA	Skip on zero or minus AC
736	SNA.SPA	Skip on non–zero and positive AC
737	SKP	Skip

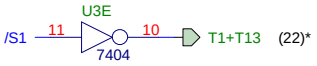
E8/T Modifications

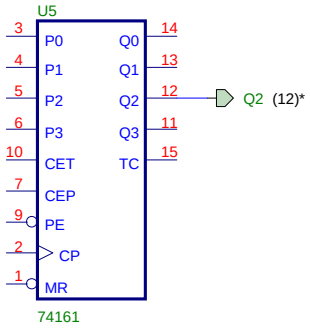
Signal labels FETCH CYC and EXECUTE CYC are incorrect.
They should be labelled T0-11 and T12-23, respectively.

Disconnect S2-9 and T12-23 from U10C.
Disconnect T14-21_2 from U15D and pin (12).
Total UL for T14-21_1 goes from 2 to 4.



Connect S1 (T1+T13) to spare pin (22).





List of New Instructions

715 DAC (decrement accumulator)

712 OSR (OR switch register with accumulator)

706 BSW (byte swap, exchange AC[7:4] with AC[3:0])

703 RLL (rotate link left through AC register)

713 CLL (clear link)

707 CML (complement link)

717 STL (set link)

731 RTL (rotate link twice left through AC register)

725 RTR (rotate link twice right through AC register)

732 SNA (skip on nonzero accumulator)

726 SPA (skip on positive accumulator)

736 SNA.SPA (skip on non-zero and positive accumulator)

723 RLR (rotate link right through AC register)

733 SZL (skip on zero link)

727 SNL (skip on nonzero link)

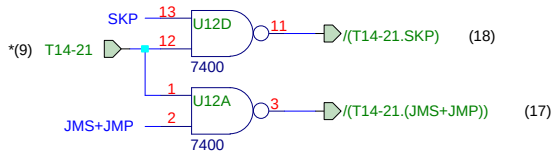
737 SKP (skip)

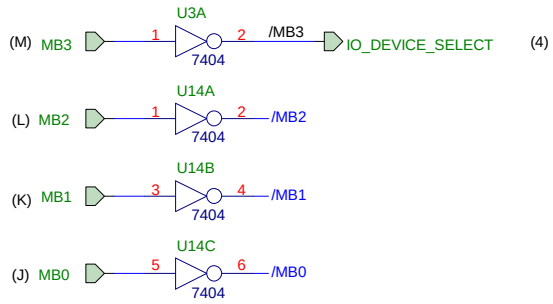
* = New Connection

E8/D Modifications

U14A, U14B, U14C and U2A are spare gates.

Disconnect T14-21-1 from (10) and connect T14-21 to T14-21-1.





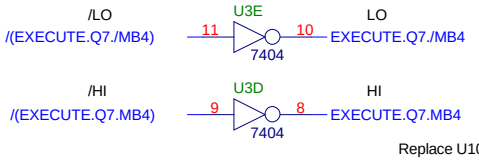
Instruction Map MB3 MB2

MB4 MB1 MB0	00	10	01	11
000	NOP	CLA	CMA	CLA.CMA
001	IAC	CLA.IAC	CMA.IAC	DAC
010	RAL	OSR	BSW	-
011	RLL	CLL	CML	STL
100	-	SZA	SMA	SZA.SMA
101	HLT	RTL	RTR	-
110	RAR	SNA	SPA	SNA.SPA
111	RLR	SZL	SNL	SKP

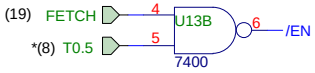
Small World Communications	Project: EDUC-8L
6 First Avenue	Macro: E8L/T & E8L/D
Payneham South SA 5070	Date: 1/22/22
Date Last Modified: 10/6/24	

E8/D Modifications (cont)

Signal labels /(Q7+EXEC.MB4) and /(Q7+EXEC).MB4 are incorrect.
They should be labelled EXECUTE.Q7./MB4 and EXECUTE.Q7.MB4.

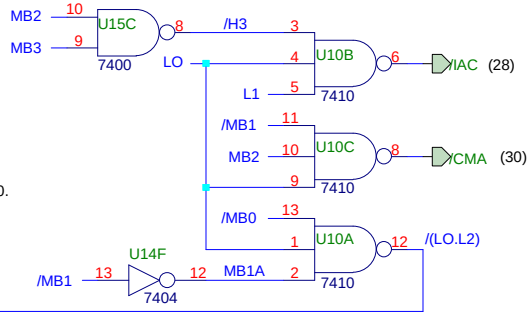


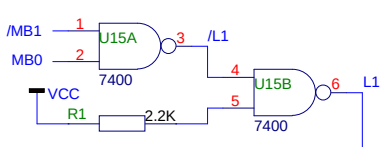
Disconnect /T0.5 and T0.5 from U14F.
Connect pin (8) to pin 5 of U13B.

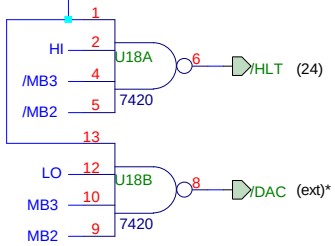


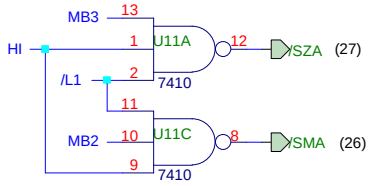
Replace U10 and U11 7400 with 7410.

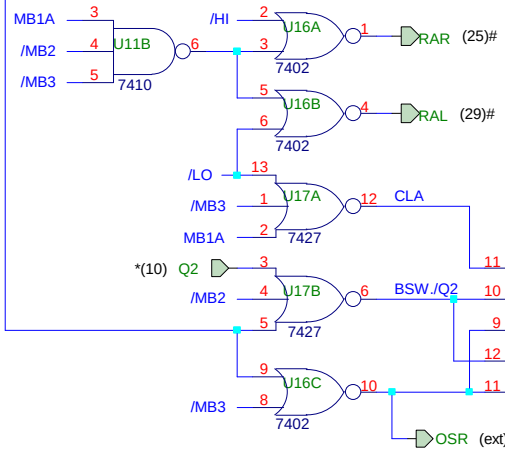
1 7400 quad 2-input NAND
1 7402 quad 2-input NOR
2 7410 triple 3-input NAND
1 7420 dual 4-input NAND
2 7427 triple 3-input NOR

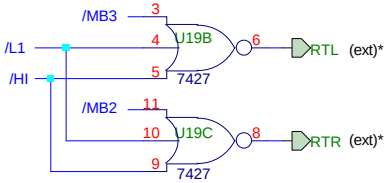


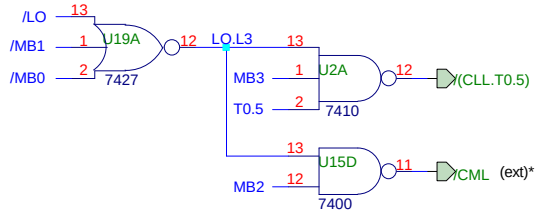












* = New Connection
\# = New Name

Instruction names are in brackets.
Instructions OSR.BSW and RTL.RTR are invalid.
STL = CLL.CML and SKP = SZL.SNL.

LO = EXEC.Q7./MB4

HI = EXEC.Q7.MB4

H0 = /MB3./MB2

H3 = MB3.MB2

L1 = /MB1.MB0

L2 = MB1./MB0

L3 = MB1.MB0

CLA = LO.MB3./MB1 (CLA, CLA.CMA, CLA.IAC, DAC)

CMA = LO.MB2./MB1 (CMA, CLA.CMA, CMA.IAC, DAC)

IAC = LO./H3.L1 (IAC, CLA.IAC, CMA.IAC)

DAC = LO.H3.L1 (DAC)

RAL = LO.H0.MB1 (RAL, RLL)

OSR = LO.MB3.L2 (OSR, OSR.BSW)

BSW = LO.MB2.L2 (BSW, OSR.BSW)

CLL = LO.MB3.L3 (CLL, CLL.CML)

CML = LO.MB2.L3 (CML, CLL.CML)

SZA = HI.MB3./L1 (SZA, SZA.SMA, SNA, SNA.SPA, SZL, SZL.SNL)

SMA = HI.MB2./L1 (SMA, SZA.SMA, SPA, SNA.SPA, SNL, SZL.SNL)

RAR = HI.H0.MB1 (RAR, RLR)

HLT = HI.H0.L1 (HLT)

RTL = HI.MB3.L1 (RTL, RTL.RTR)

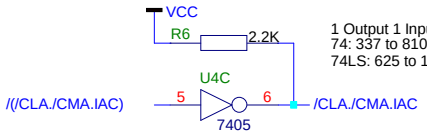
RTR = HI.MB2.L1 (RTR, RTL.RTR)

Small World Communications	Project: EDUC-8L
6 First Avenue	Macro: E8L/D
Payneham South SA 5070	Date: 1/22/22
Date Last Modified: 10/6/24	

E8/A Modifications

U4B (7405), U4C (7405), U5C (7401) and U6C (7404) are spare gates.

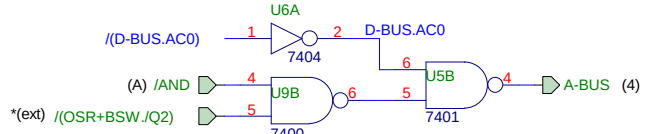
Disconnect /(/CLA./CMA.IAC) and /CLA./CMA.IAC from U10C.
Replace U10C with spare gate U4C.



1 Output 1 Input
74: 337 to 8103 ohm
74LS: 625 to 17083 ohm

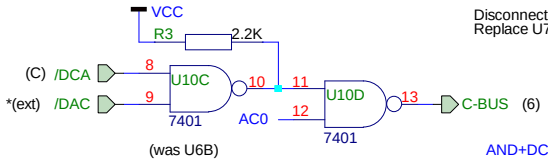
(was U10C)

Disconnect /AND from U6A.
Disconnect /(D-BUS.AC0) from U9B.



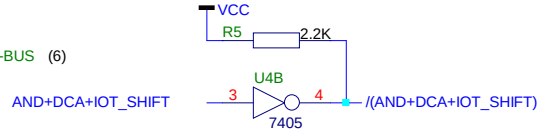
Disconnect AC7 from U1.P3
Disconnect T1 from U1./CP2 and U3./CP2.

Disconnect /DCA and DCA from U6B.
Replace U6B with U10C.



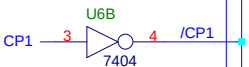
(was U6B)

Disconnect /(AND+DCA+IOT_SHIFT) and AND+DCA+IOT_SHIFT from U7B.
Replace U7B with spare gate U4B.



(was U7B)

Disconnect CP1 and /CP1 from U7C.
Replace U7C with U6B.



(was U7C)

Pin (D) renamed from /RAR to RAR.
Disconnect pin (D) from pin 12 of U7D.

Pin (8) renamed from /(T13.RAR) to ROTATE.
Disconnect pin (8) from pin 1 of U8A.

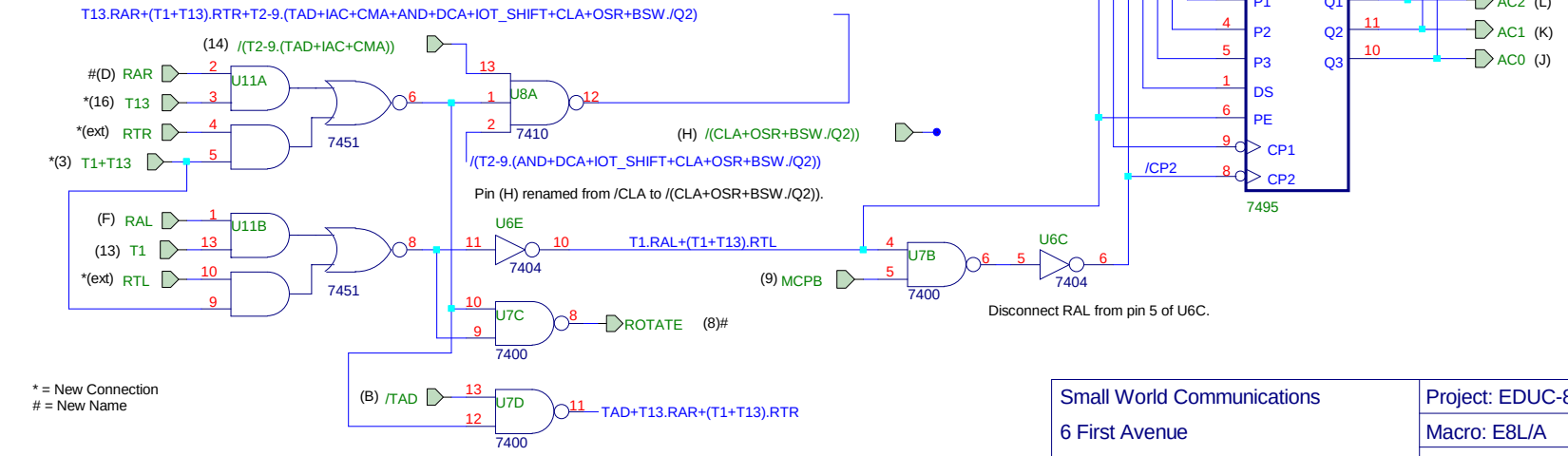
Pin (F) renamed from /RAL to RAL.
Disconnect pin (F) from pin 11 of U6E.

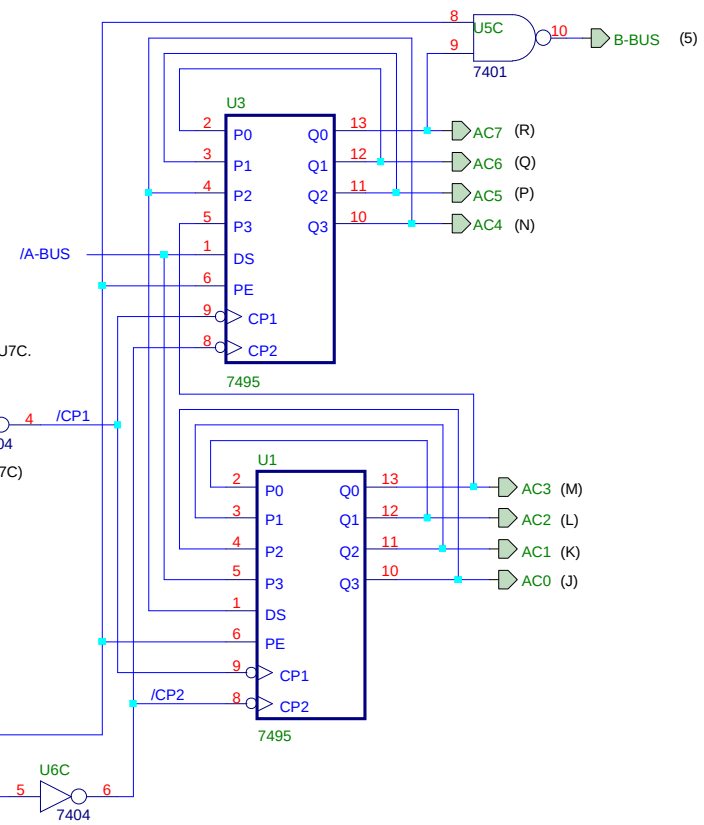
Disconnect (16) from GND and (3) from VCC.

Parts List

1x7451 dual 2-2 AOI
2x2.2K resistors

For 74LS51 connect pins 11 and 12 to VCC.





* = New Connection
\# = New Name

Small World Communications	Project: EDUC-8L
6 First Avenue	Macro: E8L/A
Payneham South SA 5070	Date: 1/22/22
Date Last Modified: 10/26/24	

E8/P Modifications

Disconnect /(T14-21.(JMP+JMS)) from U12E.

U7C and U12E are spare gates. U4B is spare FF.

For SZA and SMA have I0A = (AC=0) and I0B = (AC=-VE).
For SNA.SPA have I0A = 0 and I0B = /(AC=0)./(AC=-VE)
For SNA./SPA or /SNA.SPA have I0A = /(AC=0) and I0B = /(AC=-VE).

Disconnect AC = -VE from U11A.
Disconnect AC = 0 from U11B.

ZA = /MB0.I0A + MB0.L
ZB = /MB0.I0B + MB0./L
ZC = MB0 + /ROTATE
RLI = MB0.ROTATE

2 Outputs 1 Input
74: 337 to 4352 ohm
74LS: 625 to 9318 ohm

VCC R1 2.2K *(Q) MB1 (16) AC=0 U18B 74136
VCC R3 2.2K (H) AC=-VE U18A 74136
Disconnect GND from (Q).

SMA HI U19A SZA U19B 7422

U15 74157
I0A I0B I1A I1B I1C I1D S E GND
ZA ZB ZC ZD

U12E 7404
#(9) ROTATE
VCC R2 2.2K HI
11.8 Loads, Rmax = 4979 ohm.

*(P) MB0

ZA SZA U11B 7400 /(SZA.ZB)
SMA ZB U11A 7400 /(SMA.ZA)
(20) /(T2-9.(F+D+E))
/(T14-21.(JMP+JMS+SKP))
(15) /(T2-9.(TAD+IAC+CMA))
(19) /(T2-9.ISZ)
(22) /(T14-21.SKP)
U13A U13B 7420 EA AC
Disconnect /(RAR.T13) (9) from U13A and U13B.

Disconnect /(T2-9.JMS), /(T2-9.(F+D+E)) and T2-9.(JMS+F+D+E) from U3A.
Replace U3A with U16B.

1 Output 1 Input
VCC R4 2.2K
/(T2-9.JMS) (20) /(T2-9.(F+D+E)) U16B 7401
T2-9.(JMS+F+D+E)

Disconnect /TAD and TAD from U9C.
Replace U9C with U16C,

VCC R5 2.2K T2-9.(TAD+DAC) U6D 7400
U16C 7401 T2-9
*(ext) /DAC
(B) /TAD
(F) /IAC
U3A 7410 TAD+IAC+DAC

Disconnect A-BUS and /A-BUS from U7D.
Replace U7D with U18D,

VCC R6 2.2K
U16A 7401 A-BUS (4)
HI U18D 74136 /A-BUS

Disconnect GND from (R).
HI U18C 74136 /LINK (R)*

U7D 7400
U7C 7400
*(ext) /CML
IL

U17A 7451 LD
(5) B-BUS
(11) MCPB
U10A 7400
U9D 7404
*(32) /(CLL.T0.5)
*(29) T10
U17B 7451
HI RLI

U4B 7474
D CP SD CD Q QB
LCP HI

Disconnect /RAR (E) from U9D.
Disconnect T13 (12) and /(T13.RAR) (9) from U10A.

SR[7:0]: 4 UL, Rmax = 14688 ohm

U20 74166
HI (ext) SR7 (ext) SR6 (ext) SR5 (28) SR4 (27) SR3 (26) SR2 (25) SR1 (24) SR0 (13) T2-9
DS P0 P1 P2 P3 P4 P5 P6 P7 PE CP1 CP2 MR Q7
GND HI

*(ext) OSR U16D 7401 A-BUS

U4A 7474
CARRY_D CARRY_CP /SET_CARRY (8) /(T0+12)
D CP SD CD Q QB
CARRY /CARRY

1 Output 1 Input
74: 337 to 8103 ohm
74LS: 625 to 17083 ohm

U9C 7404
MCPB /MCPB

1x7401 quad 2-input NAND OC
1x7422 dual 4-input NAND OC
1x7451 dual 2-2 AOI
1x74136 quad 2-input XOR OC
1x74157 quad 2-input multiplexer
1x74166 8-bit parallel to serial converter

5x2.2K resistors

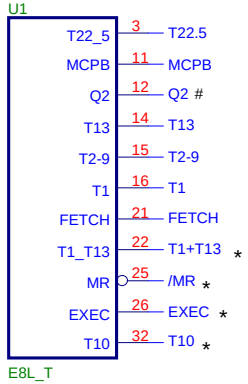
For 74LS use R1 = 0R.
For 74LS51 connect pins 11 and 12 to VCC.

* = New Connection
\# = New Name

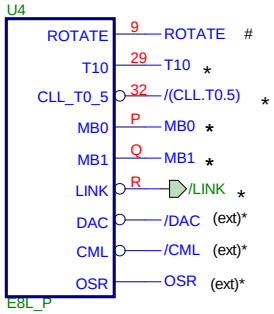
Small World Communications	Project: EDUC-8L
6 First Avenue	Macro: E8L/P
Payneham South SA 5070	Date: 1/22/22
Date Last Modified: 10/22/24	

E8/C Modifications

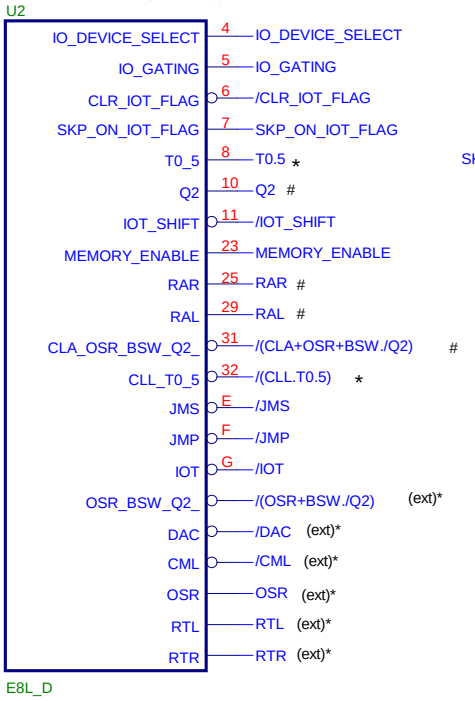
Rename T14-21 for pin E8/T(12) to Q2.
Disconnect /DEFER from pin E8/T(22).
Disconnect GND from pin E8/T(32).



Disconnect GND from pin E8/P(32).
Disconnect /RAR from pin E8/P(E).

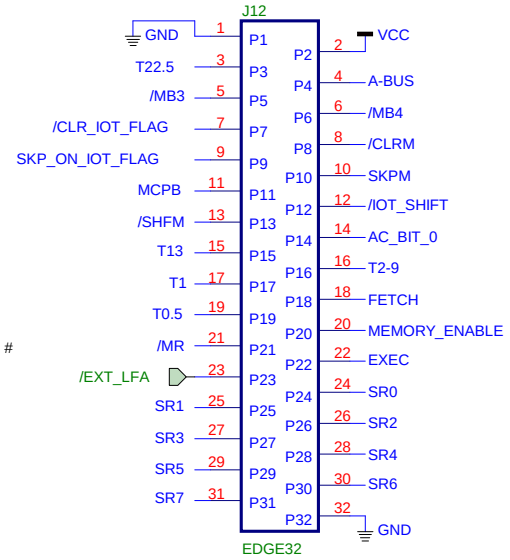


Disconnect T14-21 from pin E8/D(10).
Disconnect GND from pin E8/D(32).
Disconnect /T0.5 from pin E8/D(8).

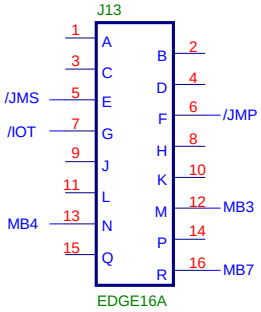


* = New Connection
\# = New Name

E8ME/B Connectors



/EXT_LFA goes to pin (11) of DB15 parallel port.

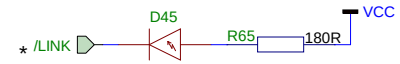


A-BUS, No=6
74 Ii = 2/2 UL: 379 to 1487 ohm
74LS Ii = 1/0.5 UL: 660 to 3203 ohm

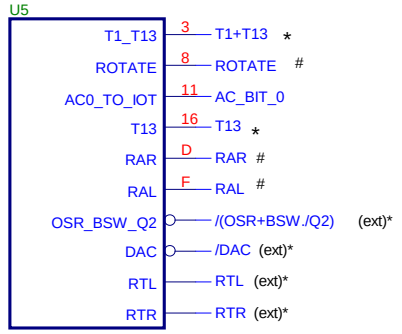
B-BUS, No=6
74 Ii = 1.4/2 UL: 379 to 1510 ohm
74LS Ii = 0.9/1.25 UL: 792 to 3223 ohm

E8/F Modifications

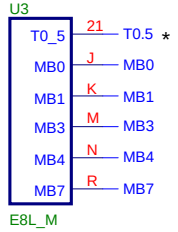
Add LED for LINK next to AC7.

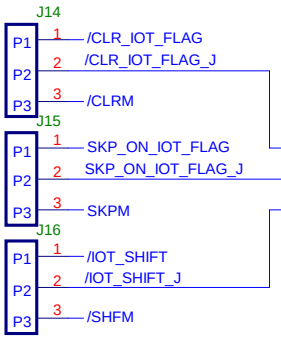


Disconnect VCC from pin E8/A(3).
Disconnect GND from pin E8/A(16).

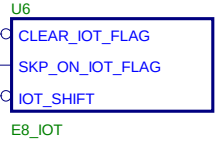


Disconnect /DEFER from pin E8/M(21).





Disconnect /CLEAR_IOT_FLAG from pin E8/IOT(7).
Disconnect SKP_ON_IOT_FLAG from pin E8/IOT(8).
Disconnect /IOT_SHIFT from pin E8/IOT(11).

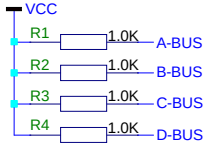


Jumpers in upper position (1-2) without EDUC-8ME.
Jumpers in lower position (2-3) with EDUC-8ME.

C-BUS, No=6
74 Ii = 2.4/3 UL: 433 to 1472 ohm
74LS Ii = 1.4/1.5 UL: 848 to 3125 ohm

D-BUS, No=2 (93415)
74 Ii = 3/3 UL: 429 to 7344 ohm
74LS Ii = 1.5/0.75 UL: 324 to 7885 ohm

D-BUS, No=1 (7401)
74 Ii = 3/3 UL: 433 to 6351 ohm
74LS Ii = 1.5/0.75 UL: 699 to 12812 ohm



Small World Communications	Project: EDUC-8L
6 First Avenue	Macro: E8L/C & E8L/F
Payneham South SA 5070	Date: 1/22/22
Date Last Modified: 10/13/24	