EE 2720

Handout # 19.

EE 2720

• Cascading priority encoders:

The subject of this handout, (the only subject), is cascading priority encoders. That is to say, implementing large priority encoders from smaller ones. I will show you one example and discuss further extensions; (that should be enough I believe). I will show you how to build a 16-to-4 priority encoder using two 8-to-3 priority encoders and some OR gates. But at first, let me expand the 8-to-3 priority encoder that we studied in handout #18. The new priority encoder presented in this handout, is an 8-to-3 priority encoder with data inputs I0, I1, ..., I7. Besides these inputs, it also has an input named EI which stands for enable input. The data outputs are Y2, Y1, Y0 like before, but besides these outputs it has two more outputs named GS (it stands for group select) and EO (it stands for enable output). Table 1, on the next page, shows the truth table of this modified or new 8-to-3 priority encoder.

→ Go to next page →

encoder disabled.

EI	I0	I1	I2	I3	I4	I5	I6	I7	Y2 Y1 Y0	GS	EO
0	X	X	X	X	X	X	X	X	0 0 0	0	0
1	X	X	X	X	X	X	X	1	1 1 1	1	0
1	X	X	X	X	X	X	1	0	1 1 0	1	0
1	X	X	X	X	X	1	0	0	1 0 1	1	0
1	X	X	X	X	1	0	0	0	1 0 0	1	0
1	X	X	X	1	0	0	0	0	0 1 1	1	0
1	X	X	1	0	0	0	0	0	0 1 0	1	0
1	X	1	0	0	0	0	0	0	0 0 1	1	0
1	1	0	0	0	0	0	0	0	0 0 0	1	0
1	0	0	0	0	0	0	0	0	0 0 0	0	1

Table 1: Truth table for an 8-to-3 priority encoder with enable input EI, enable output EO and group select output GS.

The symbol for this priority encoder of table 1 is shown in figure 1 on the next page.

→ Go to next page →

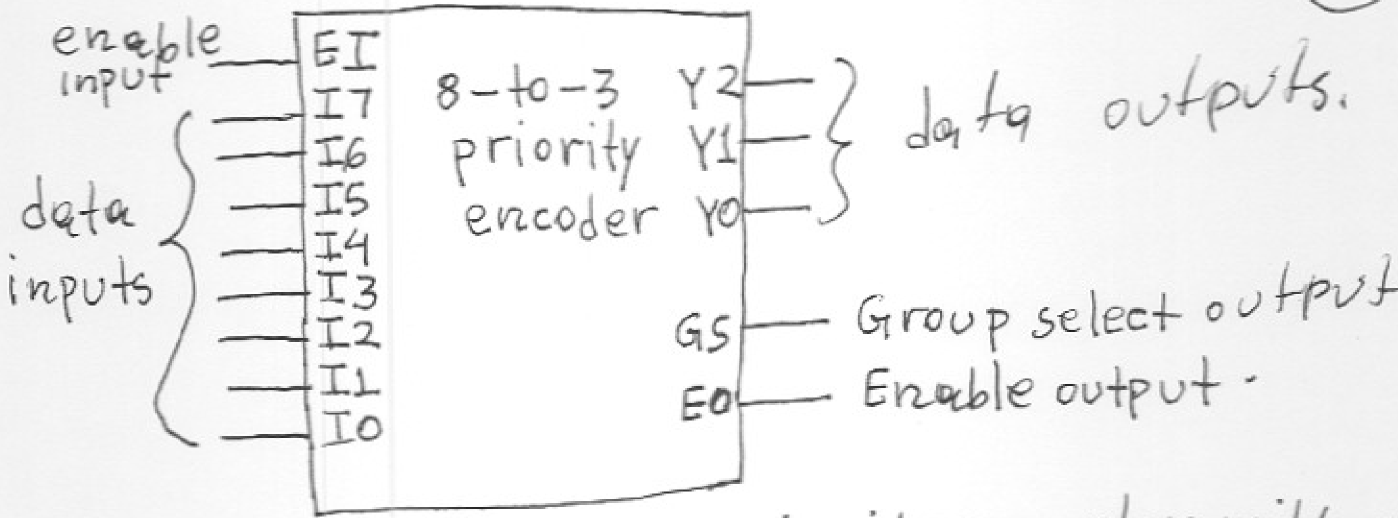

Figure 1: The 8-to-3 priority encoder with extra input EI and extra outputs GS and EO.

Let me now explain this new (modified) 8-to-3 priority encoder. The data input I7 is the highest priority input, while I0 is the lowest priority input; (I explained this in handout # 18). Furthermore, if EI=0, then the encoder is disabled which means that all its outputs become 0's. If EI=1, then the encoder is enabled and acts like a priority encoder; (see outputs Y2, Y1, Y0 on the truth table). Regarding the output GS, this output becomes 1 if the encoder is enabled and some input between I0, I1, ..., I7 is 1. otherwise, if all inputs I0, I1, .., I7 are 0's, GS=0. In this case where all inputs I0, I1, ..., I7 are 0's the outputs Y2, Y1, Y0 are also 0's. Regarding now the output EO, this output becomes 1 only if all inputs I0, I1, ..., I7 are 0's and the encoder is enabled; (EI=1). otherwise EO=0. As you will see, this output EO is useful when cascading (connecting together) priority encoders. I now present the 16-to-4 priority encoder constructed from two 8-to-3 priority encoders and some OR gates. This is shown in fig. 2 below; (some explanations will follow).

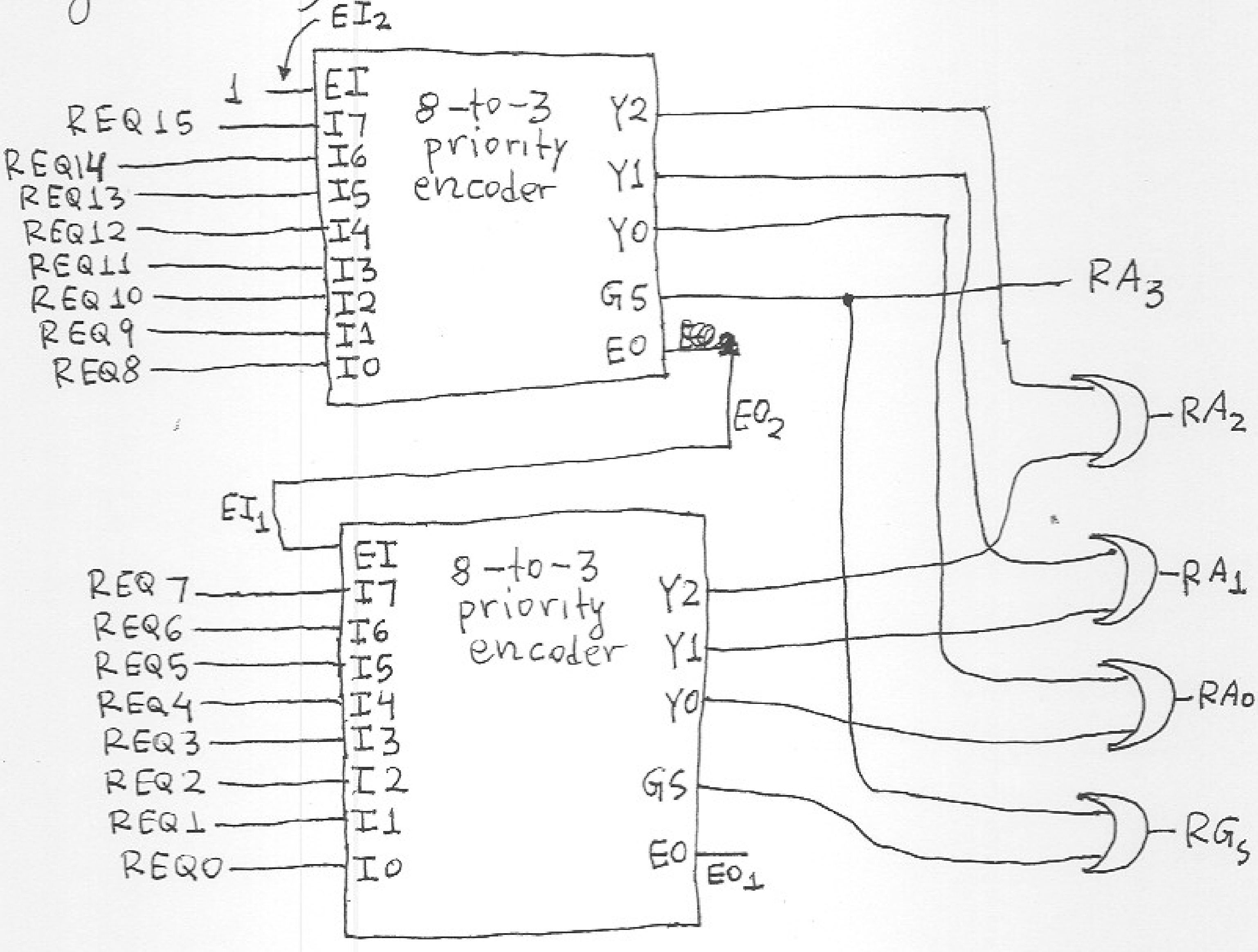

Figure 2: A 16-to-4 priority encoder constructed from two 8-to-3 priority encoders and some OR gates; (4 OR gates).

The truth table of the 16-to-4 priority encoder of figure 2 is shown in table 2 on the next page; (it doesn't fit on this page).
I now provide brief explanations about the priority encoder of figure 2.
The data inputs to the 16-to-4 priority encoder of figure 2 are REQ15, REQ14, ..., REQ1, REQ0. The input REQ15 is the highest priority input, while REQ0 is the lowest priority input; (you know what this means from handout #18). The data outputs of the 16-to-4 priority encoder are RA_3, RA_2, RA_1, RA_0. The output RA_3 is the MSB while RA_0 is the LSB. The enable input to the 16-to-4 priority encoder is EI_2 which is the enable input EI of the top 8-to-3 priority encoder. As you see, I connected this input to 1 and this way I enabled the 16-to-4 encoder. The enable output of the 16-to-4 priority encoder is EO_1 which is the enable output EO of the bottom 8-to-3 priority encoder. The group select output of the 16-to-4 priority encoder is RGs which is the result of ORING the group select outputs of the two 8-to-3 priority encoders. This way, if any of the two 8-to-3 priority encoders has a data input REQi with value 1, RGs will be 1. If all data inputs are 0s, then of course RGs will be 0. The way the 16-to-4 priority encoder operates is the following:

→ Explanations continued on page 7 →

(6).

EI_2	I_0 (RE Q_0)	I_1 (RE Q_1)	I_2 (RE Q_2)	I_3 (RE Q_3)	I_4 (RE Q_4)	I_5 (RE Q_5)	I_6 (RE Q_6)	I_7 (RE Q_7)	I_8 (RE Q_8)	I_9 (RE Q_9)	I_{10} (RE Q_{10})	I_{11} (RE Q_{11})	I_{12} (RE Q_{12})	I_{13} (RE Q_{13})	I_{14} (RE Q_{14})	I_{15} (RE Q_{15})	RA_3 RA_2 RA_1 RA_0	(E_{O2} = E_{I1})	E_{O1}	RGS
0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0000	0	0	0
1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	1111	0	0	1
1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	0	1110	0	0	1
1	X	X	X	X	X	X	X	X	X	X	X	X	X	1	0	0	1101	0	0	1
1	X	X	X	X	X	X	X	X	X	X	X	X	1	0	0	0	1100	0	0	1
1	X	X	X	X	X	X	X	X	X	X	X	1	0	0	0	0	1011	0	0	1
1	X	X	X	X	X	X	X	X	X	X	1	0	0	0	0	0	1010	0	0	1
1	X	X	X	X	X	X	X	X	X	1	0	0	0	0	0	0	1001	0	0	1
1	X	X	X	X	X	X	X	X	1	0	0	0	0	0	0	0	1000	0	0	1
1	X	X	X	X	X	X	X	1	0	0	0	0	0	0	0	0	0111	1	0	1
1	X	X	X	X	X	X	1	0	0	0	0	0	0	0	0	0	0110	1	0	1
1	X	X	X	X	X	1	0	0	0	0	0	0	0	0	0	0	0101	1	0	1
1	X	X	X	X	1	0	0	0	0	0	0	0	0	0	0	0	0100	1	0	1
1	X	X	X	1	0	0	0	0	0	0	0	0	0	0	0	0	0011	1	0	1
1	X	X	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0010	1	0	1
1	X	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0001	1	0	1
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000	1	0	1
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000	1	1	0

Table 2: Truth table for the 16-to-4 priority encoder of figure 2.

	Re q0	Re q1	Re q2	Re q3	Re q4	Re q5	Re q6	Re q7	Re q8	Re q9	Re q10	Re q11	Re q12	Re q13	Re q14	Re q15	R A	R A	R A	R A	EO2 =		R G
EI2	I0	I1	I2	I3	I4	I5	I6	I7	I8	I9	I10	I11	I12	I13	I14	I15	3	2	1	0	EI1	E0'	S
0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0	0	0
1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	1	1	1	1	0	0	1
1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	0	1	1	1	0	0	0	1
1	X	X	X	X	X	X	X	X	X	X	X	X	X	1	0	0	1	1	0	1	0	0	1
1	X	X	X	X	X	X	X	X	X	X	X	X	1	0	0	0	1	1	0	0	0	0	1
1	X	X	X	X	X	X	X	X	X	X	X	1	0	0	0	0	1	0	1	1	0	0	1
1	X	X	X	X	X	X	X	X	X	X	1	0	0	0	0	0	1	0	1	0	0	0	1
1	X	X	X	X	X	X	X	X	X	1	0	0	0	0	0	0	1	0	0	1	0	0	1
1	X	X	X	X	X	X	X	X	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1
1	X	X	X	X	X	X	X	1	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1
1	X	X	X	X	X	X	1	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1
1	X	X	X	X	X	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	1
1	X	X	X	X	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1
1	X	X	X	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1
1	X	X	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1
1	X	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

Table 2

Logic Table for16 To 4 Encoder

If one of the data inputs to the top 8-to-3 priority encoder (named REQ15, REQ14, ..., REQ8) is 1, then its enable output EO_2 becomes 0, (see table 1), and this disables the bottom 8-to-3 priority encoder; (just see in fig. 2 that the enable output of the top 8-to-3 priority encoder is connected to the enable input of the bottom 8-to-3 priority encoder). In this case, all the outputs of the bottom 8-to-3 priority encoder become 0's no matter what its data inputs are. Also, in this case, the group select output GS of the top 8-to-3 priority encoder becomes 1 which means that output line RA_3 becomes 1. The output line RA_3 is the MSB of the data output of the 16-to-4 priority encoder and it has a weight of 8; (it represents number 8). Recall that when one of the data inputs REQ_{15}, REQ_{14}, ..., REQ_8 to the top 8-to-3 priority encoder is 1, we need to represent at the outputs RA_3, RA_2, RA_1, RA_0 one of the numbers 15, 14, ..., 9, 8 all of which contain 8. Suppose now, for example, that the highest priority data input REQ15 = 1. Then we need to represent at the data outputs RA_3, RA_2, RA_1, RA_0 of the 16-to-4 encoder the number 15, or it must be $RA_3 RA_2 RA_1 RA_0$ = 1111. ~~Sin~~ This must happen no matter what the other data inputs are, because they all have lower priority than REQ15). Since REQ15 is input to line I7 of the top 8-to-3 priority encoder, this encoder will produce at it data output

Y2Y1Y0 the number 7 or Y2Y1Y0 = 111, no matter what the inputs REQ14, REQ13, ..., REQ9, REQ8 are. But we already said that $RA_3 = 1$. So the data output of the 16-to-4 priority encoder will be in this case $RA_3RA_2RA_1RA_0 = 1111 = 15_{10}$. As another example, consider that REQ15 = REQ14 = REQ13 = 0 and REQ12 = 1. Then we need to represent at the data outputs of the 16-to-4 priority encoder the number 12 (no matter what the other data inputs with lower priority than REQ12 are). Since REQ12 is input to line I4 of the top 8-to-3 priority encoder, this encoder will produce at it data output Y2Y1Y0 the number 4 or Y2Y1Y0 = 100. ~~Betewe~~ These signal Y3,Y1,Y0 will pass through the top most three OR gates unaffected, (the other inputs of these OR gates are from the bottom 8-to-3 priority encoder and are 0's), so $(RA_2, RA_1, RA_0) = 1,0,0$. But we already said that $RA_3 = 1$. So the data output of the 16-to-4 priority encoder is $RA_3 RA_2 RA_1 RA_0 = 1100 = 12_{10}$. Suppose now that all the data inputs REQ15, REQ14, ..., REQ8 to the top 8-to-3 priority encoder are 0's. Then ~~is~~ this ~~dee~~ encoder's enable output EO_2 becomes 1, (see table 1), and this obviously enables the bottom 8-to-3 priority encoder. Also, in this case, the data outputs of the top 8-to-3 priority encoder namely Y2, Y1, Y0 become 0's as well as its group select output GS becomes 0, which implies $RA_3 = 0$. Now the bottom 8-to-3 priority encoder gets enabled and works like a priority encoder. Its operation is obvious by now I guess. For example if REQ7=1, no matter what REQ6, REQ5, ..., REQ0 are, its data output is $Y_2Y_1Y_0=111$, so $RA_2RA_1RA_0=111$, if REQ7 = REQ6 = 0 and REQ5=1 no matter what REQ4, REQ3, REQ2, REQ1, REQ0 are, its data output is $Y_2Y_1Y_0=101$, so $RA_2RA_1RA_0=101$, if ~~[illegible]~~ REQ7=REQ6=REQ5=REQ4=REQ3=REQ2=0 and REQ1=1, no matter what REQ0 is, then its data output is $Y_2Y_1Y_0=001$ and so $RA_2RA_1RA_0=001$, if REQ7=REQ6=... = REQ1=0 and REQ0=1 then its data output is $Y_2Y_1Y_0$ =000 and so $RA_2RA_1RA_0=000$ etc. I hope that the way the 16-to-4 priority encoder of fig. 2 works is very clear by now. If not, ask me in class.

<u>Note</u>: You can cascade (connect together) two 16-to-4 priority encoders of fig. 2, to build a 32-to-5 priority encoder. Do it as a HW problem. By now it is very easy. I might assign it as HW problem in the next HW (not to be returned and graded most probably).

<u>Note</u>: I think I covered everything possible on encoders. So the next handout is going to be on another topic. It will be the topic of multiplexers, and demultiplexers.