3-Way Comparator
Assume normal register orientation, were the LSB (least significant bit) is on the right end.
Flow table for right-to-left information flow is:

(a)

|  | AB 00 | 01 | 11 | 10 | y1 | y2 |
|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 1 | 3 | 0 | 0 |
| 2 | 2 | 2 | 2 | 3 | 0 | 1 |
| 3 | 3 | 2 | 3 | 3 | 1 | 1 |

Flow table for left-to-right information flow is:

(b)

|  | AB 00 | 01 | 11 | 10 | y1 | y2 |
|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 1 | 3 | 0 | 0 |
| 2 | 2 | 2 | 2 | 2 | 0 | 1 |
| 3 | 3 | 3 | 3 | 3 | 1 | 1 |

NT-1. Linear iterative circuit realization.

For table (a), with the given state assignment, we have $Y1=A\bar{B}+Ay1+\bar{B}y1$,
$Y2=\bar{A}B+A\bar{B}+y2$
There is only one set of outputs for the circuit, and these are generated in the leftmost cell. The outputs are L (A<B), E, (A=B), and G (A>B). If the next-state signal in this last cell is 1, then E=1, if the next-state signal is 2, then L=1, and, if the next=state signal is 3, then G=1. The resulting expressions can be derived in a manner similar to the way we find the Y's. Taking into account the unused y-state 10, which we exploit as corresponding to a don't care row, we can see that $(\bar{A}\bar{B}+AB)\bar{y}2$ captures the 1-entries.
Note that $\bar{A}\bar{B}+AB=\overline{A\oplus B}$, so we have $E=(\overline{A\oplus B})\bar{y}2$.
In a similar manner we get $L=\bar{A}B+\bar{A}\bar{y}1y2+B\bar{y}1y2$ and $G=A\bar{B}+\bar{B}y1+Ay1$.

For table (b) we find $Y1=A\bar{B}\bar{y}2+y1$, $Y2=\bar{A}B+A\bar{B}+y2$
$E=(\overline{A\oplus B})\bar{y}2$, $L=\bar{A}B\bar{y}1+\bar{y}1y2$ and $G=A\bar{B}\bar{y}2\ \bar{y}2+y1$.

Of course, with different state assignments, we would get different (perhaps better or worse) logic expressions.

NT-2.
Assume that we use table (a), i.e., signal flow starts at the LSB end and terminates at the MSB end.
Then the complete set of mappings, followed by the multiplication table is:

| M0 | M1 | M2 |
|---|---|---|
| 1 | 2 | 3 |
| 2 | 2 | 3 |
| 3 | 2 | 3 |

| | | left | | |
|---|---|---|---|---|
| | | M0 | M1 | M2 |
| right | M0 | M0 | M1 | M2 |
| | M1 | M1 | M1 | M2 |
| | M2 | M2 | M1 | M2 |

Suppose the mapping for the entire sequence is M2, corresponding to the 10 input. That would mean that, assuming the initial state corresponding to row-1, which is should be, then starting from the MSB end and moving to the right, the first time that A and B signals differ would have to be a 10 input, i.e., where A>B. Clearly this means that the number A is greater than the number B, so G=1. If the overall mapping is M, then A=B in every position, so we should set E=1. Finally, if the overall output mapping is 01, then A<B so we should set L=1.
The tree network efficiently generates the overall mapping, and that is all we need. The outputs can be taken from the output of the root node.

A 1-hot code works nicely for the mappings, as shown below:

| mapping | w1 | w2 | w3 |
|---|---|---|---|
| M0 | 1 | 0 | 0 |
| M1 | 0 | 1 | 0 |
| M2 | 0 | 0 | 1 |

The logic for the primary level cells is obtained from

| AB | | | |
|---|---|---|---|
| 00 | 01 | 11 | 10 |
| M0 | M1 | M0 | M2 |

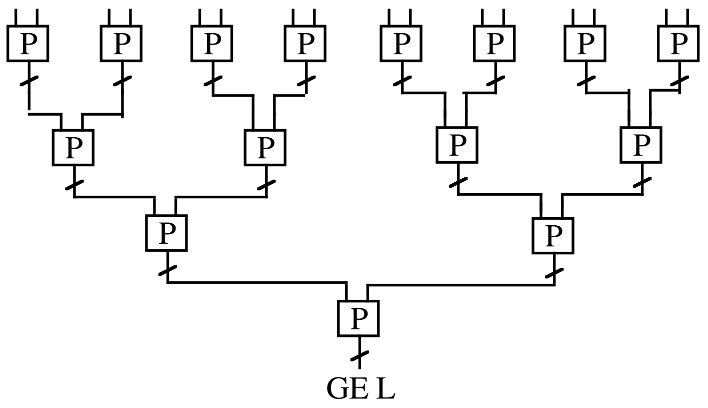
| AB | | | |
|---|---|---|---|
| 00 | 01 | 11 | 10 |
| 100 | 010 | 100 | 001 |

W1W2W3

$W1=(\overline{A\oplus B})$, $W2=\bar{A}B$, $W3=A\bar{B}$

At the root node, set E=W1, L=W2, G=W3.

The interior cell logic can be generated from the multiplication table, replacing the Mi's by the corresponding codes. We get, W1=WL1WR1, W2=WL2+WL1WR2, W3=WL3+WL1WR3



NT-3. Use flow table (a). 1-hot code state assignment as below.

| | AB | | | | | | |
|---|---|---|---|---|---|---|---|
| | 00 | 01 | 11 | 10 | y1 | y2 | y3 |
| 1 | 1 | 2 | 1 | 3 | 1 | 0 | 0 |
| 2 | 2 | 2 | 2 | 3 | 0 | 1 | 0 |
| 3 | 3 | 2 | 3 | 3 | 0 | 0 | 1 |

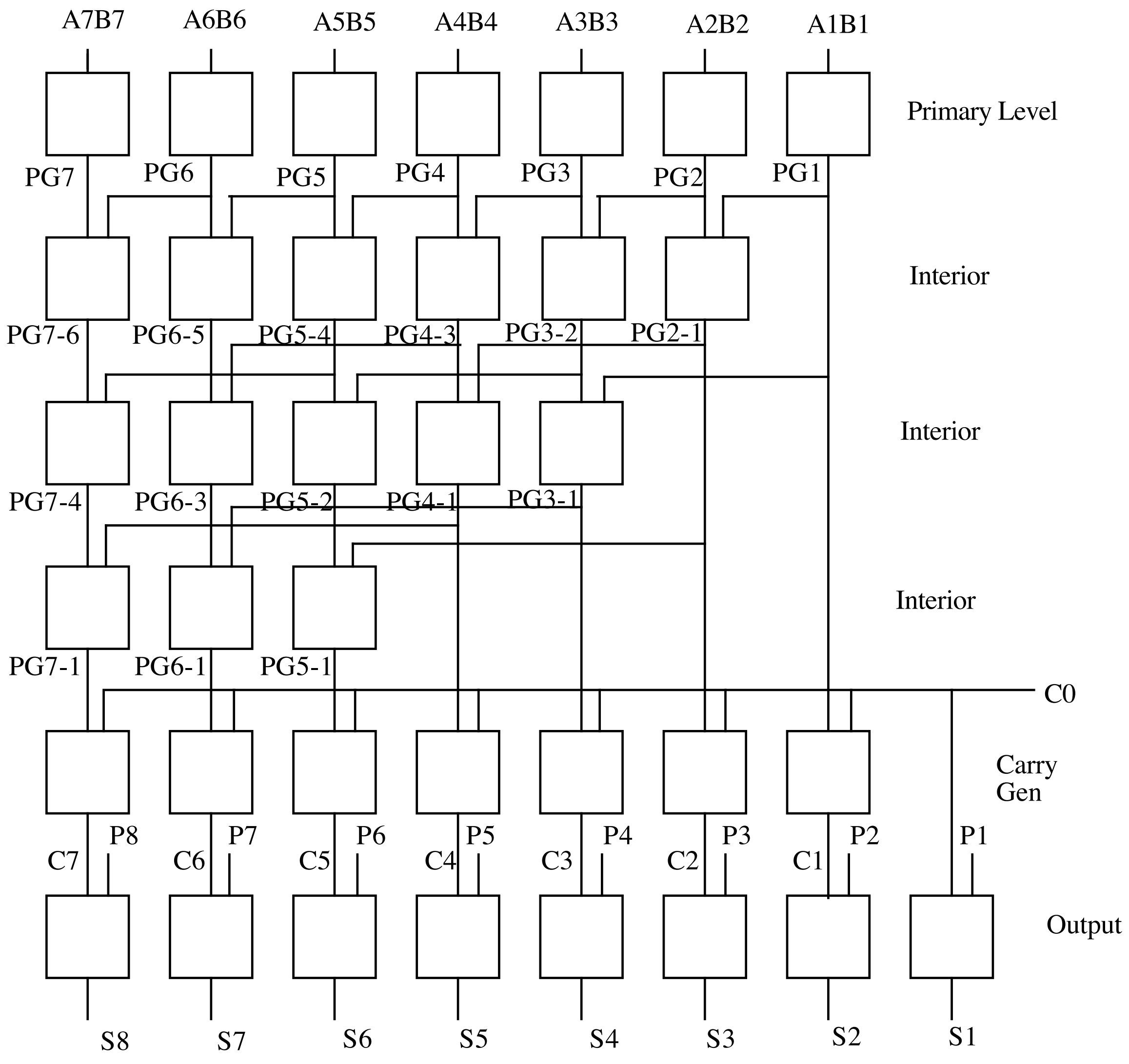
For dual-rail signals let X=1 be represented by X1, and X=0 by X0.

We will use dual-rail inputs, 1-hot code state assignment (as above), and, in effect, a 1-hot code output, with signals E, L, and G.

From the flow matrix we get the Y-logic. We need Y1=1 wherever the next-state entry is 1. These entries are both in row-1 (where y1=1), in columns 00 and 11. So we have Y1=A0B0y1+A1B1y1.

Similarly, Y2 should be 1 wherever the NS entry is 2. So we get
Y2=A0B1+A0y2+B1y2.
Finally, Y3=A1B0+A1y3+B0y3

4. Show the design of a Brent-Kung adder for 8-bit numbers. Draw a block diagram with all the primary level, interior and output generating modules. Specify logic expressions for each type of module.



The primary level modules produce the G and P signals corresponding to the input pairs. The logic is: $P_i=A_i \oplus B_i$, $G_i=A_iB_i$.

Interior level modules generate G and P signals for the inputs that they span. The logic is:
$P=P_LP_R$, $G=G_L+P_LG_R$

The carry generate modules, produce the carry signals from the P, G, and C signals they receive. The logic is: Cout=G+PCin. Note, each produces Ci from PGi-1 and C0.

The output modules produce the S signals given the input and carry signals. The logic is: $S_i=P_i \oplus C_{i-1}$. (Note that the Pi signals are themselves produced by the primary modules.)