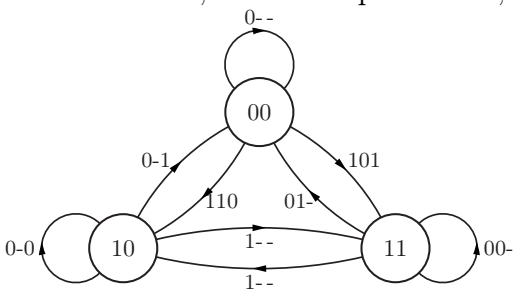
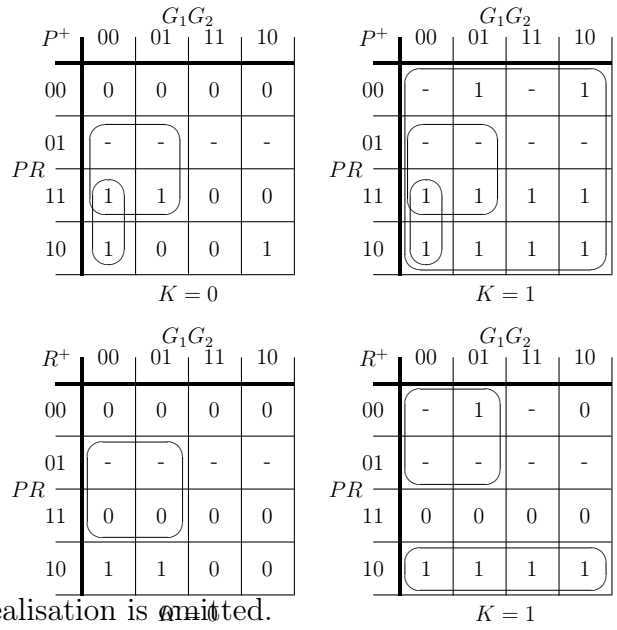
## Uppgift 3

In this problem we start by drawing a graph. It seems that it is convenient to make it as a Moore graph, since then we can let the state encoding directly control the engine. Hence as state variables we use are `P` and `R`. The state 00 means then that the door is in one of its end positions (either open or closed), 11 that the door is opening, and 10 that the door is closing. In the following graph we start in state 00, and the inputs are `K`, $\texttt{G}_1$, and $\texttt{G}_2$:




Then we get the following Karnaugh maps and (minimal) functions:




$$P^+ = K \vee RG_1' \vee PG_1'G_2'$$

$$R^+ = K'RG_1' \vee KPR' \vee KP'G_1'$$

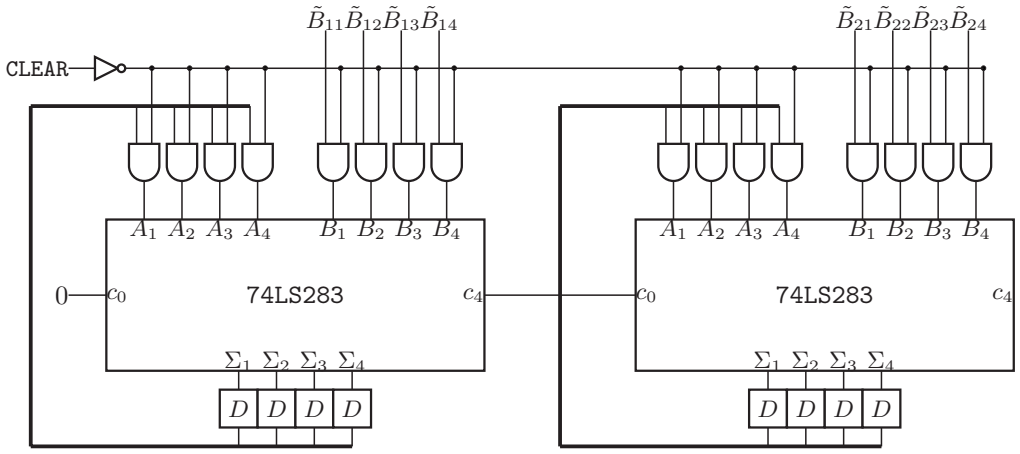
The realisation is omitted.

## Uppgift 4

Since the interval for the numbers should be $[-128, 127]$ we can use eight bit numbers (in 2-complement representation). Therefore, we cascade two of the adders 74LS283 o be able to calculate within the interval. The sum from the aders are fed to eight D-elemnets. This represents the state of the counter. The outputs from the D-elemnts are then given as input for one of the numbers for the adder, se figure below. To realise the CLEAR-signal we use the following truth table

| $x$ | `CLEAR` | $y$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

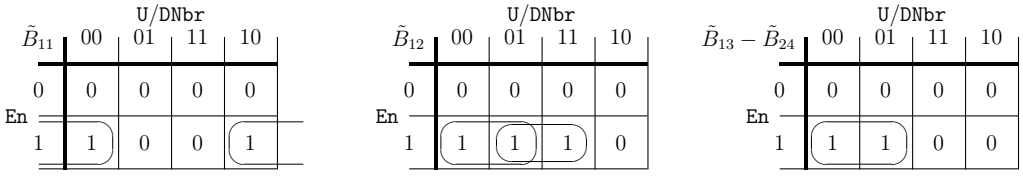
where $y = x$ if $\texttt{CLEAR} = 0$ and $y = 0$ if $\texttt{CLEAR} = 1$. This is realised by $y = x \wedge \texttt{CLEAR}'$. Putting one of these on each of the inputs for $A_i$ and $B_i$ solves the function. So, without taking the steps of the counter into consideration gives the following construction:




In the figure $\tilde{B}_{ij}$ is the $j$th input for the $i$th adder, before the CLEAR-function, where $\tilde{B}_{11}$ is the least significant bit (lsb). To find the rest of the realisation we consider the following truth table:

| En | U/D | Nbr | function | $\tilde{B}_{11}$ | $\tilde{B}_{12}$ | $\tilde{B}_{13}$ | $\tilde{B}_{14}$ | $\tilde{B}_{21}$ | $\tilde{B}_{22}$ | $\tilde{B}_{23}$ | $\tilde{B}_{24}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | – | – | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | −1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | −2 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | +1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | +2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

We see that $\tilde{B}_{13}$ to $\tilde{B}_{24}$ are realised by the same function. Hence, we have three different functions, and we draw three different Karnaugh maps to get them:

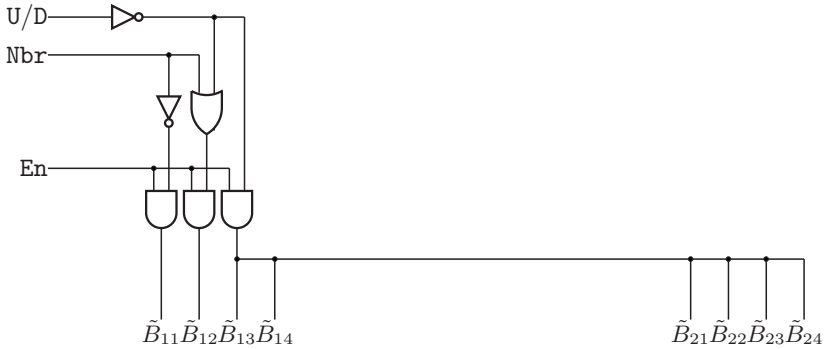



$\texttt{En} \wedge \texttt{Nbr}'$

$\texttt{En} \wedge \texttt{U/D}' \vee \texttt{En} \wedge \texttt{Nbr}$
$= \texttt{En}(\texttt{U/D}' \vee \texttt{Nbr})$

$\texttt{En} \wedge \texttt{U/D}'$

So, we complete the figure above by adding the one below.



## Uppgift 5

a) The circuit in the data sheet has three gates delay to generate the carry. If we instead cascade four FA as in the book, the first FA has three gates daeay. The rest has the delay from carry to carry of two gates. This will result in a total delay of 9 gates. Hence the solution in the data sheet is three times faster.

b) From the figure we have that

$$\Sigma_1 = (A_1 \vee B_1)(A_1' \vee B_1') \oplus C_0 = (A_1B_1' \vee A_1'B_1) \oplus C_0 = A_1 \oplus B_1 \oplus C_0$$
$$C_1 = \Big((A_1 \vee B_1)' \vee (A_1B_1)'C_0'\Big)' = (A_1 \vee B_1)(A_1B_1 \vee C_0) = A_1B_1 \vee A_1C_0 \vee B_1C_0$$

Writing these functions in a tabular, we see that the statement in the problem is true:

| $A_1B_1C_0$ | $C_1\Sigma_1$ | $2C_1+\Sigma_1$ |
|---|---|---|
| 0 0 0 | 0 0 | 0 |
| 0 0 1 | 0 1 | 1 |
| 0 1 0 | 0 1 | 1 |
| 0 1 1 | 1 0 | 2 |
| 1 0 0 | 0 1 | 1 |
| 1 0 1 | 1 0 | 2 |
| 1 1 0 | 1 0 | 2 |
| 1 1 1 | 1 1 | 3 |

c) From b) we know that we can write the $i$th carry as

$$C_i = \Big(C_{i-1}'(A_iB_i)' \vee (A_i \vee B_i)'\Big)' \tag{1}$$

At the mark for $C_2$ we have

$$\begin{aligned} C_2 &= \Big(C_0'(A_1B_1)'(A_2B_2)' \vee (A_2B_2)'(A_1 \vee B_1)' \vee (A_2 \vee B_2)'\Big)' \\ &= \Big(\underbrace{(C_0'(A_1B_1)' \vee (A_1 \vee B_1)')}_{C_1'}(A_2B_2) \vee (A_2 \vee B_2)'\Big)' = \Big(C_1'(A_2B_2) \vee (A_2 \vee B_2)'\Big)' \end{aligned}$$

and we see that $C_2$ fullfills the equation. Similarly, we can check that $C_3$ and $C_4$ satisfy the recursion in (1),

$$\begin{aligned} C_3 &= \Big(C_0'(A_1B_1)'(A_2B_2)'(A_3B_3)' \vee (A_2B_2)'(A_3B_3)'(A_1 \vee B_1)' \\ &\qquad \vee (A_3B_3)'(A_2 \vee B_2)' \vee (A_3 \vee B_3)'\Big)' \\ &= \Big(C_2'(A_3B_3)' \vee (A_3 \vee B_3)'\Big)' \end{aligned}$$

$$\begin{aligned} C_4 &= \Big(C_0'(A_1B_1)'(A_2B_2)'(A_3B_3)'(A_4B_4)' \vee (A_2B_2)'(A_3B_3)'(A_4B_4)'(A_1 \vee B_1)' \\ &\qquad \vee (A_3B_3)'(A_4B_4)'(A_2 \vee B_2)' \vee (A_4B_4)'(A_3 \vee B_3)' \vee (A_4 \vee B_4)'\Big)' \\ &= \Big(C_3'(A_4B_4)' \vee (A_4 \vee B_4)'\Big)' \end{aligned}$$

d) The recursion in (1) can be used to get a general expression for the $i$th carry. To get rid of the outer parentasis and inversion, we consider the inverse:

$$\begin{aligned}
C_i' &= C_{i-1}'(A_iB_i)' \vee (A_i \vee B_i)' \\
&= \Big(C_{i-2}'(A_{i-1}B_{i-1})' \vee (A_{i-1} \vee B_{i-1})'\Big)(A_iB_i)' \vee (A_i \vee B_i)' \\
&= C_{i-2}'(A_{i-1}B_{i-1})'(A_iB_i)' \vee (A_{i-1} \vee B_{i-1})'(A_iB_i)' \vee (A_i \vee B_i)' \\
&= \Big(C_{i-3}'(A_{i-2}B_{i-2})' \vee (A_{i-2} \vee B_{i-2})'\Big)(A_{i-1}B_{i-1})'(A_iB_i)' \\
&\qquad \vee (A_{i-1} \vee B_{i-1})'(A_iB_i)' \vee (A_i \vee B_i)' \\
&= C_{i-3}'(A_{i-2}B_{i-2})'(A_{i-1}B_{i-1})'(A_iB_i)' \vee (A_{i-2} \vee B_{i-2})'(A_{i-1}B_{i-1})'(A_iB_i)' \\
&\qquad \vee (A_{i-1} \vee B_{i-1})'(A_iB_i)' \vee (A_i \vee B_i)' \\
&= \ldots = \\
&= C_0' \bigwedge_{j=1}^{i} (A_jB_j)' \bigvee_{k=1}^{i-1} \Big((A_k \vee B_k)' \bigwedge_{\ell=k+1}^{i} (A_\ell B_\ell)'\Big) \vee (A_i \vee B_i)'
\end{aligned}$$

Hence,

$$C_i = \Big(C_0' \bigwedge_{j=1}^{i} (A_jB_j)' \bigvee_{k=1}^{i-1} \Big((A_k \vee B_k)' \bigwedge_{\ell=k+1}^{i} (A_\ell B_\ell)'\Big) \vee (A_i \vee B_i)'\Big)'$$