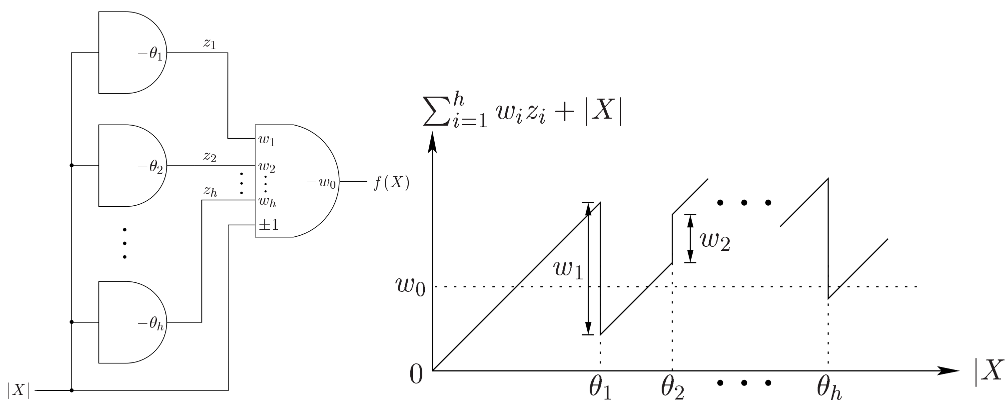
CNS/CS/EE 188a Computation Theory and Neural Systems - Homework #6

Ling Li, ling@cs.caltech.edu

March 13, 2001

6.1 Assume we only consider depth-2 circuits with the first layer consisting of TH gates.[1] Based on analysis below, Figure 1(a) gives the general yet simplified circuit that implements $f(X)$.



(a) the circuit

(b) an output pattern

Figure 1: A depth-2 circuit with TH gates in the first layer and one possible output pattern.

- **There is only one LT gate in the second layer**, since the circuit is depth-2 and there is only one output. Call that output gate O.
- WLOG, assume **all the gates in the first layer are in** $\{TH_i\}_{i=0}^{n}$. Since $\overline{TH}_i = 1 - TH_i$, if any gate in the first layer is $\overline{TH}_i$, we can change it to TH_i by also changing the corresponding weight w_i in O to $-w_i$ and subtracting the threshold w_0 by w_i. Thus the output of the changed circuit is unchanged:

$$\begin{aligned} f(X) &= \operatorname{sgn}\left(S_i + (-w_i)TH_i - (w_0 - w_i)\right) \\ &= \operatorname{sgn}\left(S_i + (-w_i)(1 - \overline{TH}_i) - w_0 + w_i\right) \\ &= \operatorname{sgn}\left(S_i + w_i\overline{TH}_i - w_0\right), \end{aligned}$$

  where S_i is the weighted sum of all the inputs except that from TH_i. Thus we can convert all $\overline{TH}_i$ gates to TH_i without changing the function of the circuit and the number of gates.

[1]If there is no restriction on the depth of the circuit, then we can use only $\lceil \log_2(n+1) \rceil$ LT gates and the gate in the first layer is also a TH gate.

- WLOG, **assume** $\theta_1 < \theta_2 < \cdots < \theta_h$. If two TH_i gates have the same threshold, we can just merge them into one gate, and sum the corresponding weights in O. This will only decrease the number of gates. Then reorder all the TH_i gates, we can make $\theta_1 < \theta_2 < \cdots < \theta_h$.

- The inputs to O include all the outputs of those TH_i gates, and x_i, the inputs to this circuit. WLOG, assume **all the weights of x_i in O is the same**. Let λ_i be the weight of x_i in O (if there is no input from x_i, $\lambda_i = 0$). WLOG, assume $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$. Let $\bar{\lambda} = \frac{1}{n}\sum_{i=1}^{n} \lambda_i$ and $S = \sum_{i=1}^{h} w_i z_i - w_0$. Since the output of a TH_i gate only depends on $|X|$, z_i does not change if $|X|$ doesn't change. So does S. Also because $f(X)$ is a symmetric function, we have for two special inputs (both with $|X| = \ell$) 1. $x_i = 1$ if $i \le \ell$, $x_i = 0$ otherwise, and 2. $x_i = 0$ if $i \le n - \ell$, and $x_i = 1$ otherwise, and any input X with $|X| = \ell$,

$$\operatorname{sgn}\left(S + \sum_{i=1}^{\ell} \lambda_i\right) = \operatorname{sgn}\left(S + \sum_{i=n-\ell+1}^{n} \lambda_i\right) = \operatorname{sgn}\left(S + \sum_{i=1}^{n} \lambda_i x_i\right). \tag{1}$$

For any input X with $|X| = \ell$, we also have

$$\sum_{i=1}^{\ell} \lambda_i \le \ell\bar{\lambda} = \sum_{i=1}^{n} \bar{\lambda} x_i \le \sum_{i=n-\ell+1}^{n} \lambda_i,$$

since we assume $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$. Thus

$$\operatorname{sgn}\left(S + \sum_{i=1}^{\ell} \lambda_i\right) \le \operatorname{sgn}\left(S + \sum_{i=1}^{n} \bar{\lambda} x_i\right) \le \operatorname{sgn}\left(S + \sum_{i=n-\ell+1}^{n} \lambda_i\right).$$

Together with (1), we have for any input X and any ℓ,

$$\operatorname{sgn}\left(S + \sum_{i=1}^{n} \bar{\lambda} x_i\right) = \operatorname{sgn}\left(S + \sum_{i=1}^{n} \lambda_i x_i\right).$$

That is, using $\bar{\lambda}$ as the weight for all x_i will not change the function of the circuit.

- Following above analysis, the inputs to O are z_i's and $|X|$, and the weight of $|X|$ is $\bar{\lambda}$. If $\bar{\lambda} > 0$, by dividing all the weights and the threshold of O by $\bar{\lambda}$, we make the weight of $|X|$ 1. Similarly, if $\bar{\lambda} < 0$, we can make it -1. *If $\bar{\lambda} = 0$, the circuit is layered and the optimal size is the number of transitions in $V(f)$ plus 1, which is no less than the number of intervals of 1's in $V(f)$ plus 1.*[2] Thus WLOG, **the weight of $|X|$ is ± 1**.

First consider the situation that the weight of $|X|$ in O is 1. Figure. 1(b) shows the weighted sum of inputs as a function of $|X|$. Basically, the sum is a straight line with slope 1, if all z_i are 0. The sum increases by w_i at $|X| = \theta_i$ since $z_i = 1$ for $|X| \ge \theta_i$. Figure. 1(b) illustrates a case where $w_1 < 0$ and $w_2 > 0$.

If the sum is above or on the line w_0, then $f(X) = 1$; otherwise $f(X) = 0$. Let c be the number of crossings between the sum line and the w_0 line. For four cases below,

[2] A trivial case is that if the number of transitions is 1, then only 1 gate is needed for both constructions.

- $v_0 = v_n = 0$. Since one crossing alternates the output $f(X)$, and the starting and ending intervals are of 0's, the number of intervals of 1's in $V(f)$ is no more than $\frac{c}{2}$, i.e., $k \le \frac{c}{2}$. It is also easy to see that $c \le 2h$. Thus $k \le h$, or the number of gates used in the circuit is $h+1 \ge k+1$.
- $v_0 = 0, v_n = 1$. Similarly, $k \le \frac{c+1}{2}$ and $c \le 2h+1$. Thus $k \le h+1$ and the number of gates is $h+1 \ge k$.
- $v_0 = 1, v_n = 0$. Now the line w_0 is below 0. $k \le \frac{c+1}{2}$ and $c \le 2h-1$. Thus $k \le h$ and the number of gates is $h+1 \ge k+1$.
- $v_0 = v_n = 1$. Similarly $k \le \frac{c}{2}+1$ and $c \le 2h$. Thus $k \le h+1$ and the number of gates is $h+1 \ge k$.

the construction in Homework #2 problem 4 is optimal. (Note that in that construction, when $v_n = 0$, we use $k+1$ gates and when $v_n = 1$, we can drop the last gate in the first layer then the number of gates is k.)

In the case that the weight of $|X|$ is -1, the inequalities between k and c remain the same for those four cases. However, the inequalities between c and h change a little. That is

- $v_0 = v_n = 0$. $k \le \frac{c}{2}$ and $c \le 2h$. Thus $h+1 \ge k+1$.
- $v_0 = 0, v_n = 1$. $k \le \frac{c+1}{2}$ and $c \le 2h-1$. Thus $h+1 \ge k+1$.
- $v_0 = 1, v_n = 0$. $k \le \frac{c+1}{2}$ and $c \le 2h+1$. Thus $h+1 \ge k$.
- $v_0 = v_n = 1$. $k \le \frac{c}{2}+1$ and $c \le 2h$. Thus $h+1 \ge k$.

Only for the case where $v_0 = 1$ and $v_n = 0$, the construction (I used) in Homework #2 problem 4 is not optimal. However, we can 'reverse' the construction to get this case right.

6.2 (a) *Product.* Look at the multiplication of two integers of $\frac{n}{2}$ bits each:

$$
\begin{array}{ccccccc}
 & & & x_2 & x_4 & \cdots & x_n \\
\times & & & x_{n-1} & \cdots & x_3 & x_1 \\
\hline
 & & & x_1 \wedge x_2 & x_1 \wedge x_4 & \cdots & x_1 \wedge x_n \\
 & & x_3 \wedge x_2 & x_3 \wedge x_4 & \cdots & x_3 \wedge x_n & \\
 & / & \vdots & \vdots & / & & \\
x_{n-1} \wedge x_2 & x_{n-1} \wedge x_4 & \cdots & x_{n-1} \wedge x_n & & & \\
\hline
\cdots & \cdots & \cdots & IP2? & \cdots & \cdots & x_1 \wedge x_n
\end{array}
$$

If there were no carries, the $\left(\frac{n}{2}-1\right)$-th bit is $(x_1 \wedge x_2) \oplus (x_3 \wedge x_4) \oplus \cdots \oplus (x_{n-1} \wedge x_n)$, i.e., $IP2(x_1, x_2, \ldots, x_n)$. However, we have to take the carries into consideration. We can insert k bits of 0's (denoted by $\mathbf{0}_k$) between every two bits in those two integers. That is, consider the multiplication of two integers of $n + (n-1)k$ bits each:

$$
\begin{array}{ccccccccc}
 & & & & x_2 & \mathbf{0}_k & \cdots & \mathbf{0}_k & x_n \\
\times & & & & x_{n-1} & \mathbf{0}_k & \cdots & \mathbf{0}_k & x_1 \\
\hline
 & & & & x_1 \wedge x_2 & \mathbf{0}_k & \cdots & \mathbf{0}_k & x_1 \wedge x_n \\
 & & / & \vdots & \vdots & \mathbf{0}_k & / & & \\
x_{n-1} \wedge x_2 & \mathbf{0}_k & \cdots & \mathbf{0}_k & x_{n-1} \wedge x_n & & & & \\
\hline
\cdots & \cdots & \cdots & \cdots & IP2 & \cdots & \cdots & \mathbf{0}_k & x_1 \wedge x_n
\end{array}
$$

The additions of (at most $\frac{n}{2}$) non-zero bits happen at column $(k+1)i$, $i = 0, 1, \ldots, n-1$ and the corresponding carries will not reach column $(k+1)(i+1)$, if $2^{k+1} > \frac{n}{2}$. Thus choose $k = \lceil \log_2(n+1) \rceil - 2$ and we can get $IP2$ by the product of two integers of length no more than $2n^2$. Since $IP2$ can not be implemented by $\widehat{LT}_2$.

(b) *Sorting*. For n even and inputs $x_1, x_2, \ldots, x_n$, construct n' integers with n' bits each, where $n' = \frac{n}{2} + 1$:

$$\begin{array}{rcl} X_0 & = & x_{n-1} \ldots x_3 x_1 0, \\ X_1 & = & x_{n-1} \ldots x_3 \bar{x}_2 1, \\ X_2 & = & x_{n-1} \ldots \bar{x}_4 x_1 1, \\ & \vdots & \\ X_{\frac{n}{2}} & = & \bar{x}_n \ldots x_3 x_1 1. \end{array}$$

That is, the least significant bit of X_0 is $X_{0,0} = 0$, and $X_{0,j} = x_{2j-1}$ for $j = 1, 2, \ldots, n'-1$; and X_i $(i = 1, 2, \ldots, n'-1)$ is just the same as X_0 except for the i^{th} bit, $X_{i,i} = \bar{x}_{2i}$, and for the least significant bit, $X_{i,0} = 1$. We have the Lemma below:

Lemma: $X_0 \neq X_i$ and, $X_0 > X_i$ if and only of $x_{2i-1} \wedge x_{2i} = 1$, where $i = 1, 2, \ldots, n'-1$.

Proof: Since $X_{0,0} = 0 \neq 1 = X_{i,0}$, $X_0 \neq X_i$. Since X_0 is the same as X_i except bit i and bit 0, we have $X_0 > X_i$ iff $(x_{2i-1} = 1) \wedge (\bar{x}_{2i} = 0)$, which is equivalent to $x_{2i-1} \wedge x_{2i} = 1$.

Feed those n' integers into a *Sorting* circuit, and we would get n' integers $X'_0, X'_1, \ldots, X'_{n'}$ out of the sorting circuit, with $X'_0 \leq X'_1 \leq \cdots \leq X'_n$. X_0 must be sorted to X'_k, where k is the number of X_i that is smaller than X_0, or by the Lemma, k is the number of 1's in $x_1 \wedge x_2, x_3 \wedge x_4, \ldots, x_{n-1} \wedge x_n$. Thus the parity of k is just the Inner-Product-Mod 2 function, i.e.,

$$IP2(x_1, x_2, \ldots, x_n) \equiv k \mod 2. \tag{2}$$

Connect the least significant bits of all X'_i with even i to a AND gate. That is, calculate $AND(X'_{0,0}, X'_{2,0}, \ldots)$. Since $X'_{i,0} = 1$ for all i except $i = k$, $AND(X'_{0,0}, X'_{2,0}, \ldots)$ is 1 if k is odd and 0 otherwise. Together with (2), we have

$$IP2(x_1, x_2, \ldots, x_n) = AND(X'_{0,0}, X'_{2,0}, \ldots).$$

Thus using a *Sorting* circuit and negative variables in the inputs and one more AND gate in the output, we implement the $IP2$ function. Suppose there is an LT gate with negative variable $\bar{x}_i$ as input. Denote the weight of $\bar{x}_i$ by w_i and the threshold of that LT gate by w_0. Then by using $-w_i$ as the weight and $w_0 + w_i$ as the threshold, we can use x_i instead of $\bar{x}_i$, since

$$w_i \bar{x}_i + w_0 = w_i(1 - x_i) + w_0 = (-w_i)x_i + (w_0 + w_i).$$

The total sum of weights (absolute values of weights) of that LT gate is at most increased by $|w_i|$. Thus we can eliminate those negative variables in the inputs and the total sum of weights is at most doubled.

To prove that $Sorting \notin \widehat{LT}_2$, we need to modify this strategy a little so that the XOR can be computed without that AND gate. However, I can not figure out the way. Sorry.

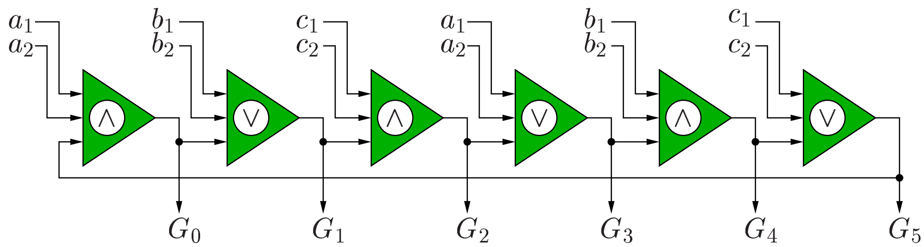


Figure 2: This combinational circuit produces six functions, each depending on all six input variables.

6.3 In this problem, $+$ is XOR, $\vee$ is OR, and $\wedge$ or just multiplication is AND.

(a) The circuit is shown in Figure 2. If we write G_i in the form of $yf_i + g_i$, where y is the value in the cyclic line, then we have $f_0 = a_1a_2$, $f_1 = (b_1 \vee b_2) + 1 = (b_1 + 1)(b_2 + 1)$, $f_2 = c_1c_2$, $f_3 = (a_1 + 1)(a_2 + 1)$, $f_4 = b_1b_2$, $f_5 = (c_1 + 1)(c_2 + 1)$. Thus

$$\prod_{i=0}^{5} f_i = a_1(a_1 + 1) \cdots c_2(c_2 + 1) \equiv 0.$$

Hence this is a combinational circuit. Then by introducing an unknown variable $y = G_5$, we get $G_0 = ya_1a_2$, $G_1 = ya_1a_2 \vee (b_1 \vee b_2)$, $G_2 = (ya_1a_2 \vee (b_1 \vee b_2)) \wedge c_1c_2$, and thus in G_3, y is eliminated:

$$\begin{aligned}
G_3 &= ((b_1 \vee b_2) \wedge c_1c_2) \vee (a_1 \vee a_2), \\
G_4 &= (c_1c_2 \vee (a_1 \vee a_2)) \wedge b_1b_2, \\
G_5 &= ((a_1 \vee a_2) \wedge b_1b_2) \vee (c_1 \vee c_2), \\
G_0 &= (b_1b_2 \vee (c_1 \vee c_2)) \wedge a_1a_2, \\
G_1 &= ((c_1 \vee c_2) \wedge a_1a_2) \vee (b_1 \vee b_2), \\
G_2 &= (a_1a_2 \vee (b_1 \vee b_2)) \wedge c_1c_2.
\end{aligned}$$

Each of these 6 functions depends on all six input variables.
If these 6 functions are implemented with an acyclic circuit, then we need at least

$$\left\lceil \frac{v-1}{d-1} \right\rceil + m - 1 = \left\lceil \frac{6-1}{3-1} \right\rceil + 6 - 1 = 8$$

fan-in 3 gates. Here $v = 6$ is the number of input variables, $d = 3$ is the fan-in, and $m = 6$ is the number of functions to be implemented.
To construct a circuit with $n(d-1)$ inputs and $2n$ fan-in d gates, for n odd and $d \geq 2$, we can expand the circuit in Figure 2. That is, suppose the $n(d-1)$ inputs are $x_{i,j}$ where $i = 0, 1, \ldots, (n-1)$ and $j = 0, 1, \ldots, (d-2)$. Gate i, which is an AND gate if i is even and an OR gate if i is odd, accepts $x_{i \bmod n, j}$ for all j and the output of gate $(i-1) \bmod 2n$ as the inputs, where $i = 0, 1, \ldots, (2n-1)$. For simplicity, the mod in the following subscripts will be omitted. (f_i and G_i always use mod $2n$ and $x_{i,j}$ uses mod n and mod $(d-1)$.)
Similar to the case $n = d = 3$, we can write the output G_i into the form $yf_i + g_i$ and find that

$$\prod_{i=0}^{2n-1} f_i = \prod_{i=0}^{n-1} \prod_{j=0}^{d-2} x_{i,j}(x_{i,j} + 1) \equiv 0.$$

In fact, we also have for any i

$$f_i \cdot f_{i+n} = \prod_{j=0}^{d-2} x_{i,j}(x_{i,j}+1) \equiv 0. \tag{3}$$

Thus this circuit is also combinational.

By symmetry, to prove each function G_i depends on all input variables, we only need to prove G_n depends on all input variables. Introduce an unknown variable $y = G_{2n-1}$, and observe the inputs and output of G_i, $i = 0, 1, \dots, n$. Gate 0 accepts y and $x_{0,j}$ for all j as inputs, and its output G_0 depends on y and all $x_{0,j}$. Gate 1 accepts G_0 and $x_{1,j}$ as inputs, and G_1 depends on G_0 and $x_{1,j}$, i.e., G_1 depends on $x_{0,j}$ and $x_{1,j}$ and y, since now y is not eliminated and $x_{1,j}$ can not eliminate any of $x_{0,j}$. Continue this analysis and we get that G_{n-1} depends on $x_{i,j}$ for all i and j, and y. As indicated in (3), G_n eliminates the effect of y (also those $x_{0,j}$ associated with y) and depends on all $x_{i,j}$ since the inputs to gate n also includes $x_{0,j}$. Thus we know all the functions G_i depend on all input variables.

To implement those $2n$ functions with an acyclic circuit, we need at least

$$\left\lceil \frac{v-1}{d-1} \right\rceil + m - 1 = \left\lceil \frac{n(d-1)-1}{d-1} \right\rceil + 2n - 1 = 3n - 1$$

fan-in d gates.

(b) Here I list all possible functions with f_i and one of a, b, or c as the inputs which depends on all 3 variables a, b, and c . If the function is just f_i itself, then it is omitted.

- $f_1 = b + abc = (\bar{a} \vee \bar{c})b$.

$$\begin{array}{ccccc} af_1 & cf_1 & \overline{af_1} & \overline{bf_1} & \overline{cf_1} \\ \hline ab\bar{c} & f_2 & f_3 & ac \vee \bar{b} & a \vee \bar{b} \vee \bar{c} \end{array}$$

- $f_2 = bc + abc = \bar{a}bc$.

$$\begin{array}{ccc} \overline{bf_2}, \overline{cf_2} & a \vee f_2 & \overline{a \vee f_2} \\ \hline a \vee \bar{b} \vee \bar{c} & f_4 & \bar{a}(\bar{b} \vee \bar{c}) \end{array}$$

- $f_3 = 1 + ab + abc = \bar{a} \vee \bar{b} \vee c$.

$$\begin{array}{ccccc} \overline{c \vee f_3} & af_3 & bf_3 & \overline{af_3} & \overline{bf_3} \\ \hline ab\bar{c} & a(\bar{b} \vee c) & (\bar{a} \vee c)b & f_7 & a\bar{c} \vee \bar{b} \end{array}$$

- $f_4 = a + bc + abc = a \vee bc$.

$$\begin{array}{ccccc} \overline{a \vee f_4} & bf_4 & cf_4 & \overline{bf_4} & \overline{cf_4} \\ \hline \bar{a}(\bar{b} \vee \bar{c}) & (a \vee c)b & f_6 & f_8 & \bar{a}\bar{b} \vee \bar{c} \end{array}$$

- $f_5 = c + bc + abc = (a \vee \bar{b})c$.

$$\begin{array}{ccccc} a \vee f_5 & \overline{a \vee f_5} & bf_5 & \overline{bf_5} & \overline{cf_5} \\ \hline a \vee \bar{b}c & \bar{a}(b \vee \bar{c}) & abc & \bar{a} \vee \bar{b} \vee \bar{c} & \bar{a}b \vee \bar{c} \end{array}$$

- $f_6 = ac + bc + abc = (a \vee b)c$.

$$\begin{array}{ccccc} a \vee f_6 & b \vee f_6 & \overline{a \vee f_6} & \overline{b \vee f_6} & \overline{cf_6} \\ \hline f_4 & ac \vee b & \bar{a}(\bar{b} \vee \bar{c}) & f_1 & \bar{a}\bar{b} \vee \bar{c} \end{array}$$

- $f_7 = 1 + a + ab + abc = \bar{a} \vee b\bar{c}$.

$$\begin{array}{cccccc} c \vee f_7 & \overline{c \vee f_7} & af_7 & bf_7 & \overline{af_7} & \overline{bf_7} \\ \hline \bar{a} \vee b \vee c & ab\bar{c} & ab\bar{c} & f_1 & f_3 & ac \vee \bar{b} \end{array}$$

- $f_8 = 1 + ab + bc + abc = \bar{a}\bar{c} \vee \bar{b}$.

$$\begin{array}{cccccc} a \vee f_8 & c \vee f_8 & \overline{a \vee f_8} & \overline{c \vee f_8} & bf_8 & \overline{bf_8} \\ \hline a \vee \bar{b} \vee \bar{c} & f_3 & f_2 & ab\bar{c} & \bar{a}b\bar{c} & a \vee \bar{b} \vee c \end{array}$$

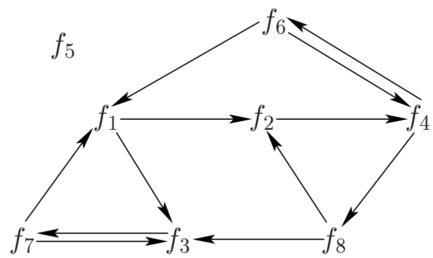


Figure 3: The function graph

Then we can make a function graph (Figure 3), with path from f_i to f_j if f_j can be expressed as a function of f_i in above. The only directed loop in the function graph is $f_1 \to f_2 \to f_4 \to f_8 \to f_3 \to f_7 \to f_1$. Thus we can draw the circuit in Figure 4. To

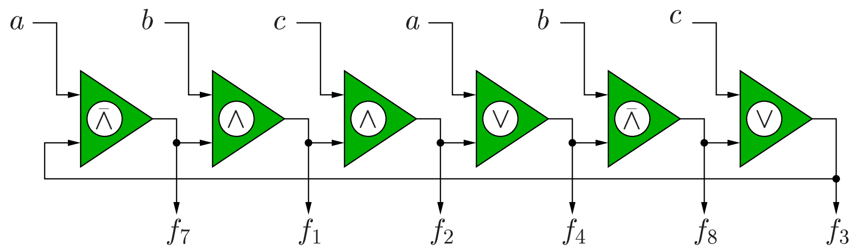


Figure 4: Circuit to implement six functions

verify the functions this circuit produces, introduce $y = f_3$. Thus $f_7 = \overline{ay} = \bar{a} \vee \bar{y}$, $f_1 = bf_7 = b(\bar{a} \vee \bar{y})$, $f_2 = cf_1 = bc(\bar{a} \vee \bar{y})$, and $f_4 = a \vee f_2 = a \vee bc$. Thus f_4 doesn't depend on y and is the same as we list above. So we know the circuit is combinational and produces correct functions as we want.

If those 6 functions are implemented with an acyclic circuit, then we need at least

$$\left\lceil \frac{v-1}{d-1} \right\rceil + m - 1 = \left\lceil \frac{3-1}{2-1} \right\rceil + 6 - 1 = 7$$

fan-in 2 gates.