1 Introduction

The procedure in the design of logical structural models with multiplexors might seem to be complete. It appears, however, that the Artjuchov-Shalyto extension of the Boolean function, which models the performance of a multiplexor, leads to its mere “setting”, and the generalised model of the multiplexor performance makes it possible to design structural models with multiplexors according to the disjoint decomposition of the given Boolean function.

2 Boolean function

Let the Boolean function
\[ f : \{0, 1\}^m \rightarrow \{0, 1\} ; \{x_1, x_2, \ldots, x_n\} \mapsto y \]
be given. If we denote the set \( \{x_i\}_{i=1}^m \) of the function \( f \) arguments by the symbol \( X \), we can write \( f(X) \) instead of \( f(x_1, x_2, \ldots, x_n) \). Let us also write \( f(x_i = \sigma_i) \) instead of \( f(x_1, x_2, \ldots, x_{i-1}, \sigma_i, x_{i+1}, \ldots, x_n) \), where \( \sigma_i \in \{0, 1\} \). We require the function \( f \) to be minimal with respect to the number of arguments, i.e., not to contain fictive arguments; the argument \( x_i \) is called fictive if \( f(x_i = 0) = f(x_i = 1) \). The term fictive if \( f(x_i = 0) = f(x_i = 1) \).

The term \( f(X) \) can be expressed by means of a canonical normal disjunctive formula – \( cndf f(X) \)

\[ f(X) = \bigvee_{\{\sigma_1, \sigma_2, \ldots, \sigma_m\} \in \{0, 1\}^m} x_1^{\sigma_1} x_2^{\sigma_2} \cdots x_m^{\sigma_m} \cdot f(\sigma_1, \sigma_2, \ldots, \sigma_m). \]

Then, if \( w_H f < 2^m/2 \) or \( w_H f > 2^m/2 \), if \( w_H f = 2^m/2 \), it is preferable to write down the respective \( cndf f(X) \) or \( cndf f(X) \), or to apply the Artjuchov-Shalyto extension of the function \( f(X) \) [1]

\[ f(X) = x_i \oplus (\bar{x}_i f(x_i = 0) \lor x_i f(x_i = 1)) \]

the validity of which can be easily confirmed by supplying 0 or 1 for \( x_i \).

Let a dichotomy \( \{X_1, X_0\} \) be given on a set of \( X \) arguments \( x_1, x_2, \ldots, x_n \) without loss of generality, such that \( X_1 = \{x_1, x_2, \ldots, x_n\} \) and \( X_0 = \{x_{n+1}, x_{n+2}, \ldots, x_m\} \), where \( n < m \). Let the simple \( k\)-multiple \((k < n)\) disjoint decomposition of the function \( f(X) \) be called the composition

\[ f(X) = \varphi(\varphi_1(X_1), \varphi_2(X_1), \ldots, \varphi_k(X_1), X_0). \]

The construction of the simple \( k\)-multiple \((k < n)\) disjoint decomposition of function \( f(X) \) can be easily done by means of a decomposition by map [2, 3].

3 Multiplexor

The term multiplexor \((MX) [2, 4, 5] \) denotes a logical object modeled both in a parametrical and an algebraic way. See Fig. 1, where \( A_r \) \((r = 1, 2, \ldots, k)\) and \( d_j \) \((j = 0, 1, \ldots, 2^k - 1)\) are the respective adjustable address- and data-input ports:

\[ y = MX(\varphi_1(X_1), \varphi_2(X_1), \ldots, \varphi_k(X_1)) \]

where

\[ g_j(X_0), g_1(X_0), \ldots, g_{2^k - 1}(X_0) \]

\[ \begin{align*}
  &\forall \varphi_1^0(X_1) \varphi_2^0(X_1) \ldots \varphi_k^0(X_1) g_j(X_0) \\
  &\forall \varphi_1^m(X_1) \varphi_2^m(X_1) \ldots \varphi_k^m(X_1) g_j(X_0)
\end{align*} \]

Fig. 1: a) Schematic diagram of a multiplexor, b) structural model of a multiplexor
where
\[ j = a_1 a_2 \ldots a_k = \sum_{r=1}^{k} a_r 2^{k-r}. \]

4 Multiplexor and the Boolean function

Let us provide the output port \( MX \) with an element of anticoincidence such that \( y = f \otimes z, \) where \( z \in \{0, 1, x, x\} \) – see Fig. 2. Let us design a multiplexor modelled by the function \( f(X) \):

\[
y = z \otimes MX(x_1, x_2, \ldots, x_m | g_0, g_1, \ldots, g_{2^n-1}) = \\
\{a_1, a_2, \ldots, a_n\}^{(01)} x_1^{a_1} x_2^{a_2} \ldots x_m^{a_n} f(a_1, a_2, \ldots, a_m).
\]

Hence \( g_j = f(a_1, a_2, \ldots, a_m) \), where

\[ j = a_1 a_2 \ldots a_m = \sum_{r=1}^{m} a_r 2^{m-r}. \]

If it is more suitable to construct \( y = cdnf f(X) \) or \( y = cnf f (X) \), see Par. 2., then the respective \( z = 0 \) or \( z = 1 \), for \( y = f(X) \oplus 0 \) or \( y = f(X) \oplus 1 \). If one cannot decide whether to construct \( f(X) \) or \( f(X) \), then if \( w_H f(X)|_{a_1, a_2, \ldots, a_n} < w_H f(X)|_{a_1, a_2, \ldots, a_n} 2^n / 2 \) or \( w_H f(X)|_{a_1, a_2, \ldots, a_n} = 1 < w_H f(X)|_{a_1, a_2, \ldots, a_n} 2^n / 2 \) then \( z = x_1 \) and

\[ y = x_1 f(x_1 = 0) \lor x_1 f(x_1 = 1) \]

or \( z = \overline{x}_1 \) and

\[ y = \overline{x}_1 f(x_1 = 0) \lor x_1 f(x_1 = 1) \]

respectively.

Example 1: Construct a multiplexor realizing the function \( f(x_1, x_2, x_3) = 0001 0111. \) Since \( w_H f = 4 = 2^3 / 2, \) as well as \( w_H f(X)|_{a_1, a_2, \ldots, a_n} = 1 < w_H f(X)|_{a_1, a_2, \ldots, a_n} 2^n / 2 \), we obtain \( z = x_1 \) and since \( f(x_1, x_2, x_3) = MX(x_1, x_2, x_3 | 0, 0, 0, 1, 0, 1, 1, 1) = \\
= x_1 \otimes MX(x_1, x_2, x_3 | d_0, d_1, d_2, d_3, d_4, d_5, d_6, d_7) \)
we obtain

\[ d_0 = d_1 = d_2 = d_3 = d_6 = d_7 = 0 \] and \( d_3 = d_4 = 1, \) i.e.

\[ y = MX(x_1, x_2, x_3 | 0, 0, 0, 1, 0, 1, 0, 0) \]
which is certainly a simpler setting of \( MX \) than that of

\[ d_0 = d_1 = d_2 = d_4 = 0 \] and \( d_3 = d_5 = d_6 = d_7 = 1. \]

5 Multiplexor and the simple k-multiple disjoint decomposition

Let \( f(x_1, x_2, \ldots, x_m) = g(x_1(X_1), x_2(X_1), \ldots, x_k(X_1), X_0) \)

where

\[ X_1 = \{x_1, x_2, \ldots, x_n\} \) and \( X_0 = \{x_{n+1}, x_{n+2}, \ldots, x_m\}, \) be a simple k-multiple \((k<n)\) disjoint decomposition of the function \( f(x_1, x_2, \ldots, x_m). \)

Let there be \( g_r = x_r (r = 1, 2, \ldots, k) \) with \( k=n \) and let us construct the arguments according to the Shannon extension of the given function

\[ f(x_1, x_2, \ldots, x_m) = \sum_{j=0}^{2^n-1} \bigvee_{r=1}^{k} x_r^{a_r} \sum_{j=0}^{2^n-1} \bigvee_{r=1}^{k} \bigvee_{j=0}^{2^n-1} g_r^{a_r} \bigvee_{j=0}^{2^n-1} g_r^{a_r} . \]

And further let

\[ f(x_1, x_2, \ldots, x_m) = MX(x_1, x_2, \ldots, x_n | g_0, g_1, \ldots, g_{2^n-1}) = \\
= \bigvee_{j=0}^{2^n-1} x_1^{a_1} \bigvee_{j=0}^{2^n-1} x_2^{a_2} \ldots \bigvee_{j=0}^{2^n-1} g_r^{a_r} \bigvee_{j=0}^{2^n-1} g_r^{a_r} ; \]

hence \( g_j = f(a_1, a_2, \ldots, a_n, x_{n+1}, x_{n+2}, \ldots, x_m) \),

where \( j = a_1 a_2 \ldots a_n = \sum_{j=1}^{m} a_r 2^{m-r}. \)

Note that the selection of arguments according to which the Shannon extension of the given function \( f(x_1, x_2, \ldots, x_m) \) is done depends completely on the view of the designer, and there is no reason to distinguish the development qualitatively according to the ‘left-side’ arguments \( x_1, x_2, \ldots, x_y \) or ‘right-side’ arguments \( x_{n+1}, x_{n+2}, \ldots, x_m \) of the function \( f \) as stated in [1].

Example 2: Let the function \( y = \bigvee (5, 6, 7, 10, 11, 19, 21, 23, 26, 27, 30, 31) \) be given; design a structural model with \( MX \) according to the Shannon development extension of the given function both according to arguments \( x_1, x_2, x_3 \) and according to arguments \( x_4, x_5, \) i.e., according to

\[ y = MX(x_1, x_2, x_3 | h_1, h_1, \ldots, h_1) = MX(x_4, x_5 | g_0, g_1, g_2, g_3) \]

Thus, let us construct decomposition maps (Fig. 3.) hence

\[
\begin{align*}
\text{a)} & \quad | x_2 & x_1 & \text{Fig. 2: Multiplexor and the element of the sum modulo 2 – M2} \\
\text{b)} & \quad | x_2 & x_1 & \text{Fig. 3: Decomposition maps of the function from Example 2}
\end{align*}
\]
Let a simple \(k\)-multiple disjoint decomposition
\[
\begin{align*}
\phi(x_1, x_2, \ldots, x_m) &= \phi(\varphi_1(x_1), \varphi_2(x_1), \ldots, \varphi_k(x_1), X_0) \\
\end{align*}
\]
be given, where \(\varphi\) will be termed an outer function and the functions \(\varphi_i\) (\(i = 1, 2, \ldots, k\)) will be denoted inner functions.

And, further, let
\[
\begin{align*}
\phi(x_1, x_2, \ldots, x_m) &= \mathcal{M}(\varphi_1(x_1), \varphi_2(x_1), \ldots, \varphi_k(x_1), X_0) \\
\end{align*}
\]

Hence
\[
\begin{align*}
\mathcal{g}_j &= \{(\varphi_1(x_1), \varphi_2(x_1), \ldots, \varphi_k(x_1), X_0) \\
\end{align*}
\]

where
\[
\begin{align*}
f = \sigma_1, \sigma_2, \ldots, \sigma_m = \sum_{i=1}^m \sigma_i 2^{m-i}. \\
\end{align*}
\]

**Example 3**: Construct a structural model with \(\mathcal{M}\) according to the decomposition

\[
\begin{align*}
\mathcal{M} &= \mathcal{M}(\varphi_1(x_1), \varphi_2(x_1), \ldots, \varphi_k(x_1), X_0) \\
\end{align*}
\]

where
\[
\begin{align*}
f = \sigma_1, \sigma_2, \ldots, \sigma_m = \sum_{i=1}^m \sigma_i 2^{m-i}. \\
\end{align*}
\]

**Fig. 4**: Structural model with \(\mathcal{M}\) from Example 2

\[
y = \overline{x_1 \overline{x_2} x_3(0)} \lor \overline{x_1 \overline{x_2} x_3 x_4 \lor x_5} \lor \overline{x_1 \overline{x_2} x_3 x_4} \\
\lor \overline{x_1 \overline{x_2} x_3 x_4} \lor x_3 \overline{x_2} x_3 x_4 \lor x_3 \lor x_3 \overline{x_2} x_3 \lor x_1 \overline{x_2} x_3 \lor x_1 \overline{x_2} x_3 \lor x_1 \overline{x_2} x_3 \lor x_1 \overline{x_2} x_3.
\]

Hence the structural models from Fig. 4. Note that in Fig. 4b) a ROM module is suggested and in Fig. 4c) the structure is realized only with multiplexer modules.

**Fig. 5**: Decomposition map of the function from Example 3

\[
y = \mathcal{M}(\varphi_1(x_1), \varphi_2(x_1), \varphi_3(x_1), X_0) \\
\]

of the function \(y\) from Example 2. According to the decomposition map (Fig. 5) we obtain

\[
\begin{align*}
\varphi_1(x_1, x_2, x_3) &= \overline{x_1 \overline{x_2} x_3} \\
\varphi_2(x_1, x_2, x_3) &= x_1 \overline{x_2} x_3 \\
\end{align*}
\]

for the inner functions.

Since
\[
y = \mathcal{M}(\varphi_1, \varphi_2, \varphi_3 | \mathcal{g}_0, \mathcal{g}_1, \mathcal{g}_2, \mathcal{g}_3)
\]

we obtain
\[
y = \overline{x_1 \overline{x_2} x_3} \lor \overline{x_1 \overline{x_2} x_3} \lor \varphi_1 \varphi_2(x_4, x_5) \lor \varphi_1 \varphi_2(x_4, x_5)
\]

and hence the structural model in Fig. 6.
6 Conclusions

The multiplexer appears to be a very helpful MSI module. The design of structural models is sufficiently simple and suitable also for the implementation of logical functions on chips provided with FPGA or FPD.

References


Doc. Ing. Josef Bokr, CSc.  
e-mail: bokr@kiv.zcu.cz  
Department of Information and Computer Science  
University of West Bohemia  
Faculty of Applied Sciences  
Universitní 22  
306 14 Pilsen, Czech Republic

Doc. Ing. Vlastimil Jáneš, GSc.  
e-mail: janes@fd.cvut.cz  
Department of Control and Telematics  
Czech Technical University in Prague  
Faculty of Transportation Sciences  
Konviktská 20  
110 00 Prague 1, Czech Republic