Домашнне задание №18
Григорьев Дмитрий БПМИ–163

Задание 1.
Решение:

Посмотрим на таблицу истинности для функции $x \oplus y \oplus z$.
Заметим, что функция истинна, если ровно один из аргументов истинен или все три сразу.
Тогда можно заменить функцию $x \oplus y \oplus z$ на $\overline{(x \wedge (y \vee z) \vee y \wedge (x \vee z) \vee z \wedge (x \vee y))} \wedge (x \vee y \vee z)$.
То есть мы получили функцию, использующую только одно отрицание, значит возможно построить необходимую схему.

ч.т.д.

Задание 2.
Решение:

Выпишем последние 9 наборов переменных. Заметим, что в 8 последних из них $x_1 = 1$, а в оставшемся девятом наборе $x_1 = 0, x_2 = 1, x_3 = 1, x_4 = 1$.
Поэтому нам достаточно построить схему для функции $x_1 \vee (x_2 \wedge x_3 \wedge x_4)$.
Схема:
$x_1, x_2, x_3, x_4, x_2 \wedge x_3, (x_2 \wedge x_3) \wedge x_4, x_1 \vee ((x_2 \wedge x_3) \wedge x_4)$

Задание 3.
Решение:

Нам на вход подается n битов. Нужно проверить, что $x_1 = 1, x_3 = 1$, а все остальные равны 0.
Схема:
$x_1, x_2, ..., x_n, \neg x_2, \neg x_4, \neg x_5, ..., \neg x_n, x_1 \wedge \neg x_2, (x_1 \wedge \neg x_2) \wedge x_3, ((x_1 \wedge \neg x_2) \wedge x_3) \wedge \neg x_4, ..., x_1 \wedge \neg x_2 \wedge x_3 \wedge \neg x_4 \wedge ... \wedge \neg x_n$

Задание 4.
Решение:

Нам нужно умножить двоичное число на 3. Пусть число длины n. Тогда нам необходимо умножить исходное число x на 2 и прибавить к результату x.
Чтобы умножить число x на 2 нужно дописать к двоичной записи x '0' справа. Т.е. $x_0 = 0$, так же для удобства запишем в y исходное число x с дописанным '0' слева, этот ноль ничего не изменит, но будет удобнее складывать 2 числа длины $n + 1$, получаем: $y_0 = x_1, y_1 = x_2, ..., y_{n-1} = x_n, y_n = 0$. Теперь нужно построить схему сложения этих 2 чисел длины $n + 1$.
Мы хотим построить схему с $2 \cdot (n + 1)$ входами и $n + 2$ выходами. Идея конструкции схемы будет та же, что и в обычном школьном сложении в столбик. Мы будем

складывать числа x и y поразрядно, попутно вычисляя биты переноса в следующий разряд. Для удобства будем обозначать через b_i бит, который переносится в i-ый разряд из предыдущих.
Введем дополнительные схемы:
•Схема для MAJ_3:
$x_1, x_2, x_3, x_1 \vee x_2, (x_1 \vee x_2) \wedge x_3, (x_1 \wedge x_2), ((x_1 \vee x_2) \wedge x_3) \vee (x_1 \wedge x_2)$
•Схема для xor от двух переменных:
$x_1, x_2, \overline{x_1}, \overline{x_2}, \overline{x_1} \wedge x_2, x_1 \wedge \overline{x_2}, (\overline{x_1} \wedge x_2) \vee (x_1 \wedge \overline{x_2})$
Будем сохранять ответ в переменную z. z_0 можно посчитать сразу $z_0 = x_0 \oplus y_0$, далее видно, что $b_1 = MAJ_3(x_0, y_0, 0)$, $z_1 = x_1 \oplus y_1 \oplus b_1$, далее для любых z_i, b_i вычисления будут такие же как и для z_1 и b_1, только в вычислении b_i заменить в MAJ_3 '0' на b_{i-1}.
Схема:
$x_0, x_1, ..., x_n, y_0, y_1, ..., y_n, b_1 = MAJ_3(x_0, y_0, 0), b_2 = MAJ_3(x_1, y_1, b_1), ..., b_n = MAJ_3(x_{n-1}, y_{n-1}, b_{n-1}),$
$x_1 \oplus y_1, ..., x_n \oplus y_n, \underline{x_0 \oplus y_0, (x_1 \oplus y_1) \oplus b_1, ..., (x_n \oplus y_n) \oplus b_n, MAJ_3(x_n, y_n, b_n).}$

$n + 2$ выхода подчеркнуты в схеме.

Задание 5.
Решение:

Нам нужно проверить, делится ли число на 3. Для этого достаточно рассмотреть знакочередующуюся сумму, а точнее ее делимость на 3(Так как мы можем воспользоваться следующим признаком делимости: Если основание системы счисления равно k - 1 по модулю некоторого числа k, то любое число делится на k тогда и только тогда, когда сумма цифр, занимающих нечётные места, либо равна сумме цифр, занимающих чётные места, либо отличается от неё на число, делящееся на k без остатка).
Теперь, для того, чтобы посчитать остаток от деления на 3 знакочередующейся суммы, мы будем последовательно идти по всем битам, начиная с младшего разряда, и поддерживать остаток от деления на 3 текущей суммы. Для этого рассмотрим четные и нечетные позиции:
•Четные позиции

Если текущий бит равен '0', то остаток не изменится, если же '1', тогда 0 перейдет в 1, 1 перейдет в 2, 2 перейдет в 0.
Теперь необходимо как то поддерживать остатки. Будем хранить их в виде двоичного числа, младший разряд которого на i-ом шаге равен a_i, а старший b_i, тогда 00, 01, 10 должно переходить в 01, 10, 00, если $x_i = 1$, и 00, 01, 10 остаться таким же, если $x_i = 0$.
Заметим, что $b_i = a_{i-1}$, а $a_i = \overline{a_{i-1} \vee b_{i-1}}$, только мы не учли, что при $x_i = 0$ мы не должны пересчитывать остаток. Получится:
$b_i = a_{i-1} \wedge x_i \vee b_{i-1} \wedge \overline{x_i}$
$a_i = \overline{a_{i-1} \vee b_{i-1}} \wedge x_i \vee a_{i-1} \wedge \overline{x_i}$

•Нечетные позиции

Если текущий бит равен '0', то остаток не изменится, если же '1', тогда 0 перейдет в 2, 1 перейдет в 0, 2 перейдет в 1.

Остатки должны пересчитаться из 00, 01, 10 в 10, 00, 01, пересчитывать будем так:
$b_i = \overline{a_{i-1} \vee b_{i-1}} \wedge x_i \vee b_{i-1} \wedge \overline{x_i}$
$a_i = b_{i-1} \wedge x_i \vee a_{i-1} \wedge \overline{x_i}$

Теперь мы научились пересчитывать остатки, нужно указать только, что $a_0 = 0, b_0 = x_0$. Ответом будет $\overline{a_n \vee b_n}$