

Программное моделирование комбинационной схемы

Необходимо реализовать программу, формирующую функцию вычисления выходов произвольной комбинационной схемы, описанной в формате json.

Пользовательский интерфейс командной строки

```
<cmd> <in.json> <out.c>
```

- `in.json` – путь к файлу с входными данными: описанием комбинационной схемы в формате json;
- `out.c` – путь, по которому программа должна сформировать компилируемый C-файл с искомой функцией `calc`.

Прототип функции `calc`

```
void calc (const uint8_t in [], uint8_t out []);
```

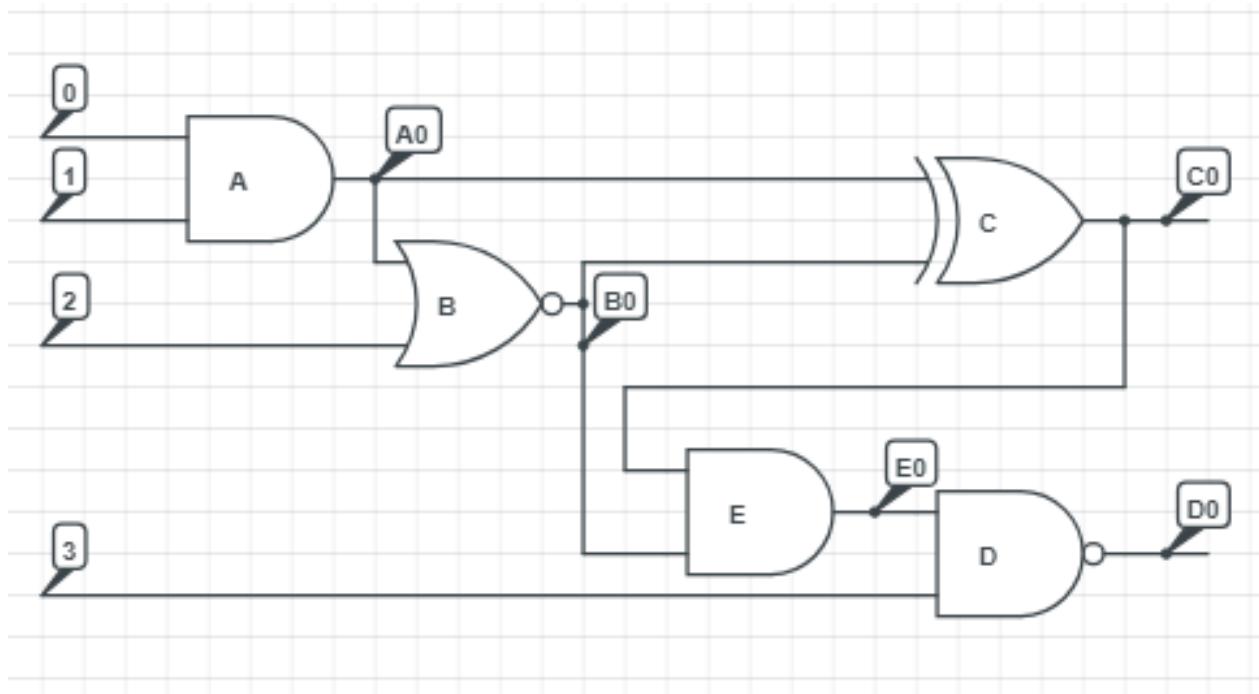
, где `in`, `out` – значения входа и выхода схемы: младший (нулевой) разряд нулевого элемента массива соответствует младшему разряду значения, первый разряд нулевого элемента массива – первому разряду значения и т.д.;

За управление памятью отвечает вызывающий контекст, размер массивов `in`, `out` достаточен для хранения входных / выходных значений схемы.

Язык реализации: любой, рекомендуется python или C++ с библиотеками boost.json .

1 Пример описания комбинационной схемы в формате json

1.1 Описанная комбинационная схема



1.2 Описание схемы в формате json

```
{  
    "gates":  
    {  
        "nor":  
        {  
            "inw": 2  
            , "outw": 1  
            , "table": [1, 0, 0, 0]  
        }  
        , "and":  
        {  
            "inw": 2  
            , "outw": 1  
            , "table": [0, 0, 0, 1]  
        }  
        , "nand":  
        {  
            "inw": 2  
            , "outw": 1  
            , "table": [1, 1, 1, 0]  
        }  
        , "xor":  
        {  
            "inw": 2  
            , "outw": 1  
            , "table": [0, 1, 1, 0]  
        }  
    }  
    , "schematics":  
    {  
        "inw": 4  
        , "outw": 2  
        , "gates":  
        {  
            "A": "and"  
            , "B": "nor"  
            , "C": "xor"  
            , "D": "nand"  
            , "E": "and"  
        }  
        , "drivers":  
        {  
            "C0": "A0"  
            , "C1": "B0"  
            , "A0": 0  
            , "A1": 1  
            , "B0": "A0"  
            , "B1": 2  
            , "D0": "E0"  
            , "D1": 3  
            , "E0": "C0"  
            , "E1": "B0"  
        }  
        , "output": ["C0", "D0"]  
    }  
}
```

1.3 Легенда

- Раздел `gates` – перечень (json-словарь) моделей библиотечных вентилей:
 - `inw/outw` – разрядности входа/выхода;
 - `table` – таблица преобразований входа в выход: `outw[k] = (table [inw] >> k) & 1.`
- Раздел `schematics` – описание цифровой схемы:
 - `inw/outw` – разрядность входа/выхода,
 - `gates` – вентили схемы (из библиотечных корневого раздела `gates`),
 - `drivers` – описание межсоединений в формате
 - | "<вентиль схемы 1><номер входа вентиля>"
: "<вентиль схемы 2><номер выхода вентиля>" |
 - `output` – массив точек выхода схемы размера `schematics.outw`.