Замечания по выполнению лабораторных работ по дисциплине “Проектирование цифровых устройств” по методичке “Лабораторный практикум по дисциплине “Схемотехника ЭВМ”, автор М.Н. Томчук”.

28 пункт в методичке, рисунок 36. В блоке OA (операционный автомат) на вход hClock провести не clk\_25mhz, а ledn[7] (выход USBBridge: OClk). Пример на рисунке 1.

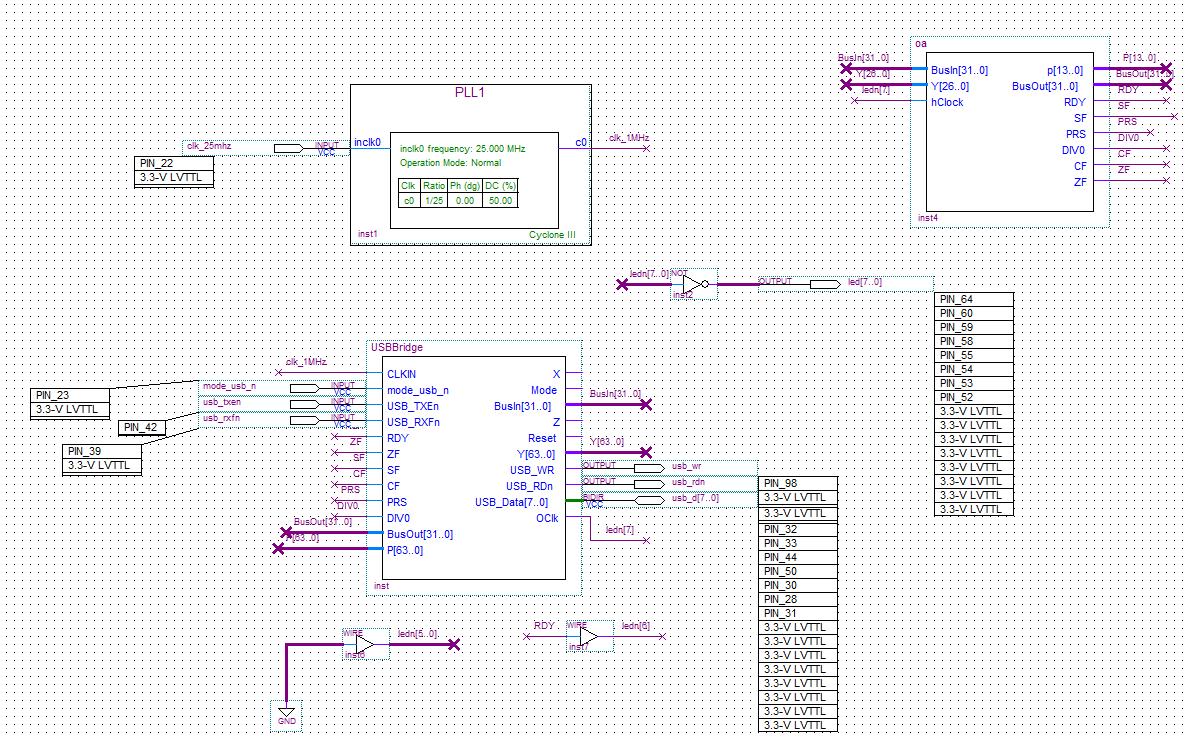


Рисунок 1 – Пример подключения выхода ledn[7] к входу hClock операционного автомата

Может возникнуть ситуация, когда в регистр записываются некорректные значения и в схеме используется управляющий сигнал в качестве входа некоторого логического элемента. В данной ситуации требуется построить схему задержки с использованием D-триггера. Управляющий сигнал необходимо записывать в D-триггер по фронту синхроимпульса(clk). Пример на рисунке 2-3.

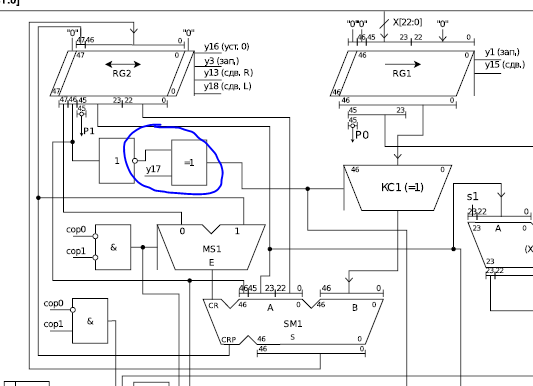


Рисунок 2 – Ситуация, требующая задержки

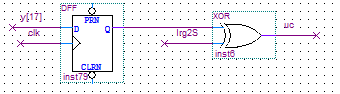


Рисунок 3 – Пример схемы задержки

Особенности работы с программой SchemKP.exe(Схемотехника ЭВМ. Синтез ОЧ АЛУ)

Установка представлена на рисунке 4.

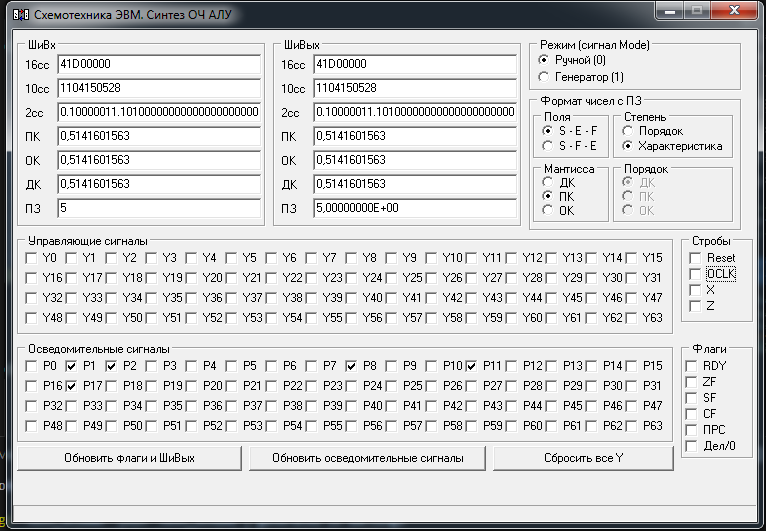


Рисунок 4 – Окно установки

Принцип работы с программой вы найдете в методичке.

Чтобы вывести результат вычислений на ШиВых, нужно установить Строб Z. Далее установить и снять OCLK два раза. При этом Z автоматически снимется.

Автоматизация работы установки.

На лабораторной работе защита состоит в демонстрации работы с помощью ручного режима, но выполнение цикла занимает большое количество времени, поэтому было найдено решение в виде автоматизации с помощью программы AutoHotKey, которая позволяет автоматизировать действия в системе Microsoft Windows. Документацию и ссылку на скачивание программы можно найти на сайте https://ahk-wiki.ru/.

Были реализованы следующие функции для автоматизации работы в установке:

1. Clock(arr): void
2. Check(p): bool

Clock(arr) принимает массив управляющих сигналов - Yi в формате:

Clock(["Y2", "Y18", "Y7"])

Данная функция устанавливает управляющие сигналы (устанавливает галочки), подает синхроимпульс, а затем снимает управляющие сигналы.

Функция Check(p) принимает осведомительный сигнал и выдает значения true или false в зависимости от его значения. (галочка – true, нет галочки – false).

Check("P14")

Реализация функций представлена ниже.

WinTitle = Схемотехника ЭВМ. Синтез ОЧ АЛУ

SetTitleMatchMode, 3

Clock(arr) {

    for index, element in arr

    {

        Set(element)

    }

    Set("OCLK")

    Sleep, 150

    UnSet("OCLK")

    for index, element in arr

    {

        UnSet(element)

    }

}

Check(p) {

    sleep, 200

    global WinTitle

    ;ControlFocus, %P%, %WinTitle%

    ControlGet, text, Checked,, %p%, %WinTitle%

    return text

}

Set(y) {

    global WinTitle

    ;ControlFocus, %y%, %WinTitle%

    Control, Check,, %y%, %WinTitle%

}

UnSet(y) {

    global WinTitle

    ;ControlFocus, %y%, %WinTitle%

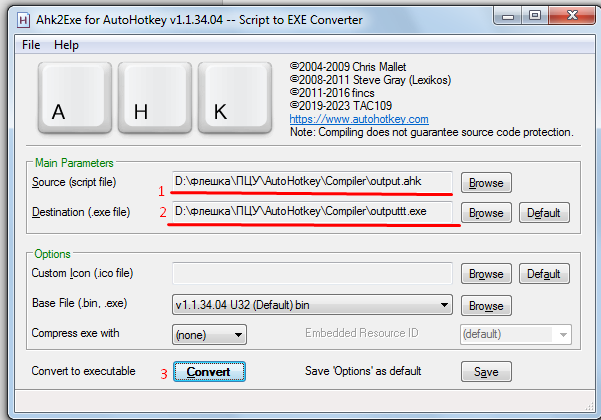
    Control, Uncheck,, %y%, %WinTitle%

}

Данный код функций следует поместить в файл, который затем следует скомпилировать в исполняемый файл.

Пример использования представлен в приложении А.

Компиляция ahk скрипта:



1 – указание пути исходного кода скрипта.

2 – указание пути выполняемого файла скрипта.

3 – кнопка компиляции.

Будет создан исполняемый файл, при запуске которого будет запущен Ahk скрипт. Чтобы выполнить определенную функцию, следует привязать её на определенную клавишу или сочетание клавиш. Например, привязка функции reset() к клавише F5.

F5::reset()

УДАЧИ!!!!!!!

Приложение А

WinTitle = Схемотехника ЭВМ. Синтез ОЧ АЛУ

SetTitleMatchMode, 3

F10::UA()

F5::reset()

F9::UA()

F6::div\_while()

F7::logic\_while()

F11::sub\_while\_1()

F12::sub\_while\_2()

UA() {

    Clock(["Y1", "Y5"])

    if Check("P14") {

        if not Check("P0") {

            fzero()

            return

        } else {

            f1isz()

            return

        }

    }

    if not Check("P0") {

        f2isz()

        return

    }

    Clock(["Y2", "Y18", "Y7"])

    if Check("P2") {

        Clock(["Y2"])

        f2isz()

        return

    }

    if Check("P3") {

        f1isz()

        return

    }

    if Check("P10") {

        Clock(["Y2"])

        f2isz()

        return

    }

    if Check("P11") {

        f1isz()

        return

    }

    Clock(["Y2"])

    while not Check("P9") {

        if Check("P8") {

            Clock(["Y9", "Y10"])

        } else {

            Clock(["Y8", "Y11"])

        }

    }

    if Check("P16") {

        Clock(["Y16", "Y3"])

    } else {

        Clock(["Y16", "Y4", "Y3"])

    }

        if Check("P14") {

        Clock(["Y23"])

        Clock(["Y63", "Y62"])

        return

    }

    if Check("P15") {

        Clock(["Y17"])

        Clock(["Y3", "Y14"])

    }

    if Check("P12") {

        Clock(["Y9", "Y10"])

        if Check("P13") {

            Clock(["Y59", "Y63"])

        } else {

            Clock(["Y61", "Y63"])

        }

        return

    }

    while not Check("P1") {

        Clock(["Y15", "Y13"])

        if Check("P13") {

            fzero()

            return

        }

    }

    if Check("P5") {

        Clock(["Y29"])

    }

    Clock(["Y63", "Y61"])

}

fzero() {

    Clock(["Y23", "Y27"])

    Clock(["Y63", "Y62"])

}

f2isz() {

    if Check("P15") {

        Clock(["Y29", "Y17"])

        Clock(["Y14", "Y3"])

    }

    Clock(["Y63", "Y61"])

}

f1isz() {

    Clock(["Y23", "Y27"])

    Clock(["Y2", "Y3"])

    if not Check("P16") {

        Clock(["Y29"])

    }

    Clock(["Y63", "Y61"])

}

reset() {

    Clock(["Y0", "Y20", "Y30", "Y27", "Y23", "Y57"])

}

DNN() {

    Clock("Y11")

}

div\_while() {

    while not Check("P6") {

        if Check("P5") {

            Clock(["Y3","Y16"])

        } else {

            Clock(["Y3","Y16","Y4"])

        }

        Clock(["Y15","Y21","Y22"])

    }

}

logic\_while() {

    while not Check("P1") {

        Clock(["Y15","Y13"])

        if Check("P13") {

            break

        }

    }

}

sub\_while\_1() {

    while not Check("P9") {

        if Check("P8") {

            Clock(["Y10", "Y9"])

        } else {

            Clock(["Y11", "Y8"])

        }

    }

}

sub\_while\_2() {

    while not Check("P1") {

        Clock(["Y15","Y13"])

        if Check("P13") {

            break

        }

    }

}

UA\_example() {

*;Clock(["Y0", "Y1", "Y2"])*

*;Clock(["Y4", "Y6"])*

    c:=0

    while not Check("P6") and c < 40  {

        Clock(["Y3"])

        Clock(["Y7"])

        c++

    }

*; if NOT Check("P0")  {*

*;     DNN()*

*;     return*

*; }*

}

Clock(arr) {

    for index, element in arr

    {

        Set(element)

    }

    Set("OCLK")

    Sleep, 150

    UnSet("OCLK")

    for index, element in arr

    {

        UnSet(element)

    }

}

Check(p) {

    sleep, 200

    global WinTitle

*;ControlFocus, %P%, %WinTitle%*

    ControlGet, text, Checked,, %p%, %WinTitle%

    return text

}

Set(y) {

    global WinTitle

*;ControlFocus, %y%, %WinTitle%*

    Control, Check,, %y%, %WinTitle%

}

UnSet(y) {

    global WinTitle

*;ControlFocus, %y%, %WinTitle%*

    Control, Uncheck,, %y%, %WinTitle%

}