

实验名称：简单中断接口电路实现

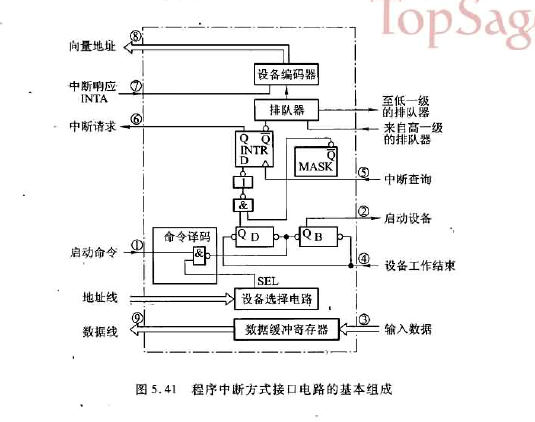
学号： 1140310606

姓名： 张茗帅

班级： 1403106

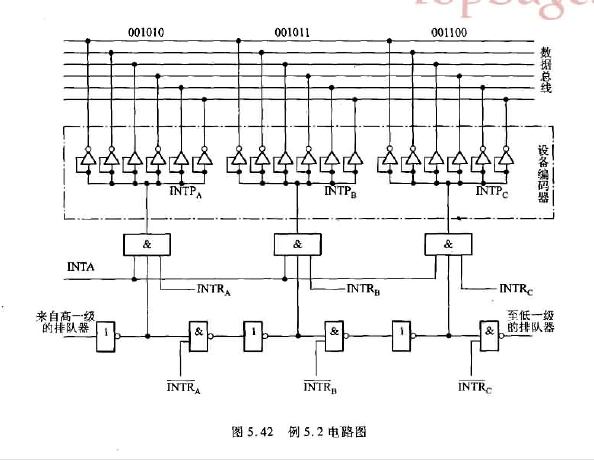
简单中断接口电路的实现

1. I/O中断处理的介绍



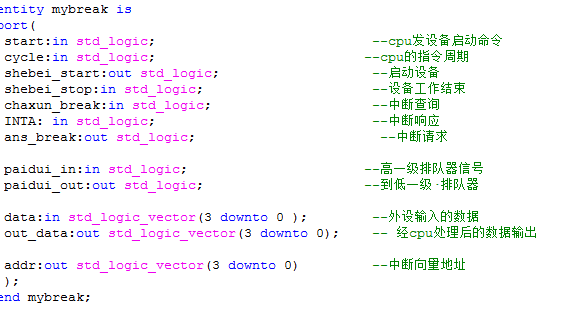
I/O处理中断过程：

1. 由cpu发I/O设备启动命令，将接口中的B置‘1’，D置‘0’。
2. 接口启动输入设备开始工作
3. 输入设备将数据送入数据缓冲寄存器
4. 输入设备向接口发出”设备工作结束“信号，将D置‘1’，B置‘0’,标志设备准备就绪
5. 当设备准备就绪（D=1），且本设备未被屏蔽（MASK=0）时，在指令执行阶段的结束时刻，由cpu发出中断查询信号
6. 设备中断请求触发器INTR被置‘1’，标志设备向cpu提出中断请求。INTR送至排队器中，进行中断判优
7. 若cpu允许中断（EINT=1），设备又被排队选中，进入中断响应阶段，而中断响应信号INTA将排队器输出送至编码器形成向量地址
8. 向量地址送至PC，作为下一条指令的地址
9. 由于向量地址中存放的是一条无条件转移指令，故这条指令执行结束后，即无条件转至该设备的服务程序入口地址，开始执行中断服务程序，进入中断服务阶段，通过输入指令将数据缓冲寄存器的输入数据送至CPU的通用寄存器，再存入相关主存单元
10. 中断服务程序的最后一条指令是中断返回指令，当其执行结束是，即中断返回至原程序的断点处。



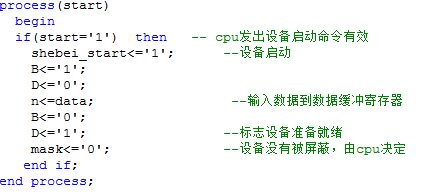
1. 代码相关模块的介绍
2. 端口设置

如下，代码注释有，就不细说

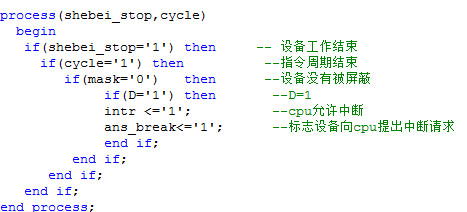


1. 操作步骤的实现

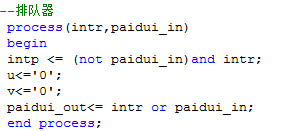
（1）处理过程的1-5步



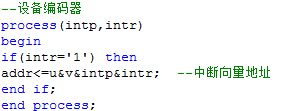
（2）第六步，在指令周期结束响应中断



（3）排队器实现的较简单

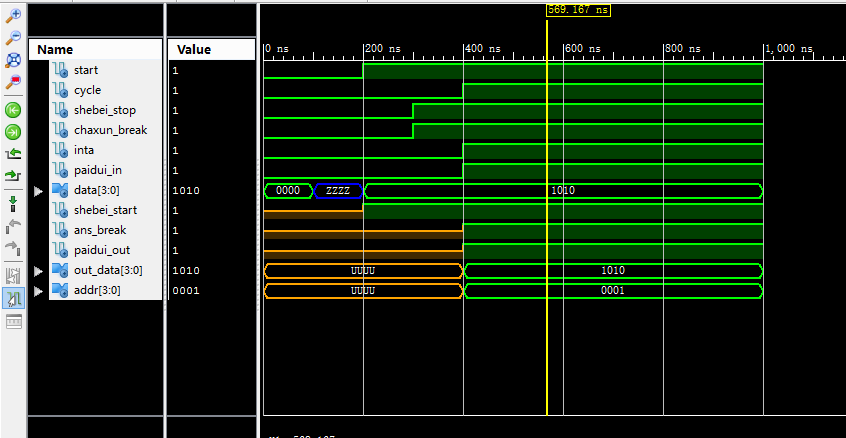


（4）设备编码器



实现的简单，得出的中断向量地址，是由一个为00和INTP INTR合成的。具体的编码，没有实现，此处仅作为一个例子

1. 仿真结果



1. 实验心得体会

本次实验较难，很多具体的细节都没能实现，整个过程可以理解到了，但是涉及到一些具体的电路实现，还不会。总之，此次实验收获很多，相比上次实验挑战性更高，学的也更多。

五、源代码

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

-- Uncomment the following library declaration if using

-- arithmetic functions with Signed or Unsigned values

--use IEEE.NUMERIC\_STD.ALL;

-- Uncomment the following library declaration if instantiating

-- any Xilinx primitives in this code.

--library UNISIM;

--use UNISIM.VComponents.all;

entity mybreak is

port(

start:in std\_logic; --cpu发设备启动命令

cycle:in std\_logic; --cpu的指令周期

shebei\_start:out std\_logic; --启动设备

shebei\_stop:in std\_logic; --设备工作结束

chaxun\_break:in std\_logic; --中断查询

INTA: in std\_logic; --中断响应

ans\_break:out std\_logic; --中断请求

paidui\_in:in std\_logic; --高一级排队器信号

paidui\_out:out std\_logic; --到低一级·排队器

data:in std\_logic\_vector(3 downto 0 ); --外设输入的数据

out\_data:out std\_logic\_vector(3 downto 0); -- 经cpu处理后的数据输出

addr:out std\_logic\_vector(3 downto 0) --中断向量地址

);

end mybreak;

architecture Behavioral of mybreak is

signal n: std\_logic\_vector(3 downto 0); --数据缓冲寄存器

signal intr:std\_logic; --允许中断信号

signal mask:std\_logic; --设备是否屏蔽信号

signal D:std\_logic;

signal B:std\_logic;

signal intp:std\_logic;

signal u:std\_logic;

signal v:std\_logic;

begin

process(start)

begin

if(start='1') then -- cpu发出设备启动命令有效

shebei\_start<='1'; --设备启动

B<='1';

D<='0';

n<=data; --输入数据到数据缓冲寄存器

B<='0';

D<='1'; --标志设备准备就绪

mask<='0'; --设备没有被屏蔽，由cpu决定

end if;

end process;

process(shebei\_stop,cycle)

begin

if(shebei\_stop='1') then -- 设备工作结束

if(cycle='1') then --指令周期结束

if(mask='0') then --设备没有被屏蔽

if(D='1') then --D=1

intr <='1'; --cpu允许中断

ans\_break<='1'; --标志设备向cpu提出中断请求

end if;

end if;

end if;

end if;

end process;

--排队器

process(intr,paidui\_in)

begin

intp <= (not paidui\_in)and intr;

u<='0';

v<='0';

paidui\_out<= intr or paidui\_in;

end process;

--设备编码器

process(intp,intr)

begin

if(intr='1') then

addr<=u&v&intp&intr; --中断向量地址

end if;

end process;

process(intr)

begin

if(intr ='1') then --cpu响应中断

out\_data<=n; -- 响应中断之后，运行中断服务程序，开始读出数据

end if;

end process;

end Behavioral;

测试代码：

ENTITY mybreak\_test5 IS

END mybreak\_test5;

ARCHITECTURE behavior OF mybreak\_test5 IS

-- Component Declaration for the Unit Under Test (UUT)

COMPONENT mybreak

PORT(

start : IN std\_logic;

cycle : IN std\_logic;

shebei\_start : OUT std\_logic;

shebei\_stop : IN std\_logic;

chaxun\_break : IN std\_logic;

INTA : IN std\_logic;

ans\_break : OUT std\_logic;

paidui\_in : IN std\_logic;

paidui\_out : OUT std\_logic;

data : IN std\_logic\_vector(3 downto 0);

out\_data : OUT std\_logic\_vector(3 downto 0);

addr : OUT std\_logic\_vector(3 downto 0)

);

END COMPONENT;

--Inputs

signal start : std\_logic := '0';

signal cycle : std\_logic := '0';

signal shebei\_stop : std\_logic := '0';

signal chaxun\_break : std\_logic := '0';

signal INTA : std\_logic := '0';

signal paidui\_in : std\_logic := '0';

signal data : std\_logic\_vector(3 downto 0) := (others => '0');

--Outputs

signal shebei\_start : std\_logic;

signal ans\_break : std\_logic;

signal paidui\_out : std\_logic;

signal out\_data : std\_logic\_vector(3 downto 0);

signal addr : std\_logic\_vector(3 downto 0);

-- No clocks detected in port list. Replace <clock> below with

-- appropriate port name

BEGIN

-- Instantiate the Unit Under Test (UUT)

uut: mybreak PORT MAP (

start => start,

cycle => cycle,

shebei\_start => shebei\_start,

shebei\_stop => shebei\_stop,

chaxun\_break => chaxun\_break,

INTA => INTA,

ans\_break => ans\_break,

paidui\_in => paidui\_in,

paidui\_out => paidui\_out,

data => data,

out\_data => out\_data,

addr => addr

);

-- Stimulus process

stim\_proc: process

begin

-- hold reset state for 100 ns.

wait for 100 ns;

data<="ZZZZ";

out\_data<="ZZZZ";

wait for 100 ns;

start<='1';

data<="1010";

wait for 100 ns;

shebei\_stop<='1'; --设备工作结束

chaxun\_break<='1';

wait for 100 ns;

INTA<='1';

cycle<='1';

paidui\_in<='1';

wait;

end process;

END;