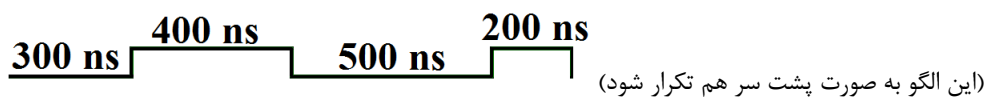
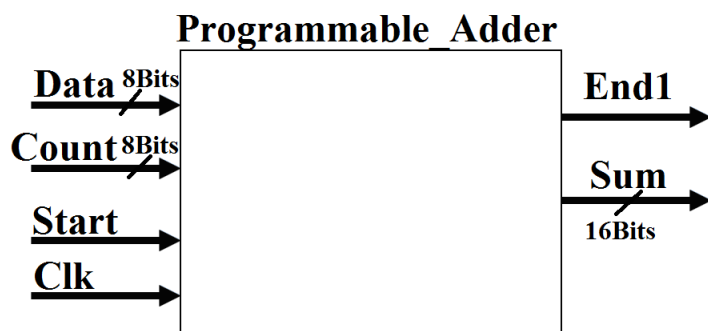


در این بخش به حل سه مساله با استفاده از ماشین حالت پرداخته خواهد شد:

- ایجاد یک شکل موج متناوب به صورت زیر، با استفاده از کلاک با فرکانس ۱۰۰ مگاهرتز (دوره تناوب ۱۰ نانوثانیه)



- طراحی یک جمع کننده قابل برنامه ریزی (قابلیت جمع کردن  $n$  عدد متوالی)



- جابجایی مقادیر داخل بلاک رم (مقادیر ذخیره شده در آدرسهای زوج و فرد یک بلاک رم یک کیلو بایتی را جابجا کند)

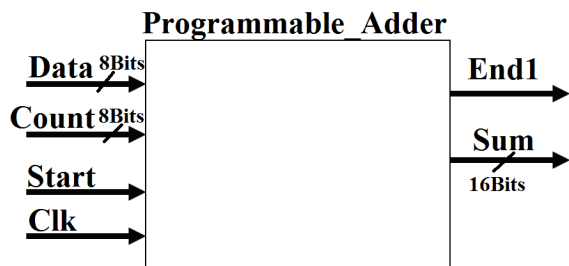
- نمونه سوال



ایجاد شکل موج متناوب فوق، شامل ۴ مرحله خواهد بود که در یک حلقه باید تکرار شوند. با توجه به اینکه کلاک ورودی دارای دوره تناوب ۱۰ نانوثانیه است بنابراین برای محاسبه ۳۰۰ نانو ثانیه تاخیر کافی است ۳۰ کلاک بشماریم و به همین ترتیب ۴۰، ۵۰ و ۲۰. بنابراین برنامه به صورت زیر خواهد شد:

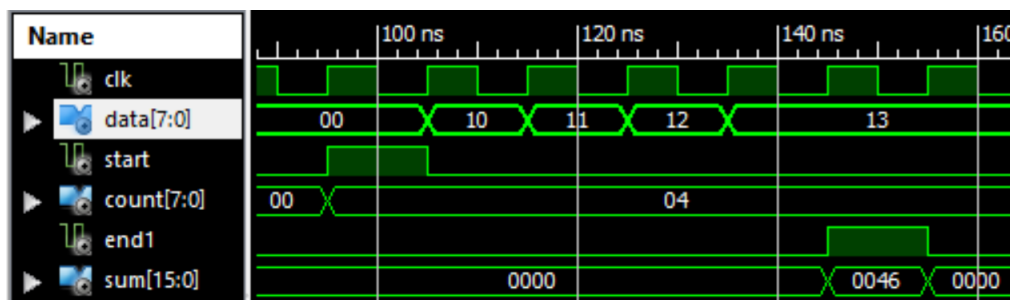
```
process (Clk)
begin
    if (Clk='1' and Clk'event) then
        Counter <= Counter + 1;
        case state is
            when S0 =>
                Outp <= '0';
                if (Counter=30) then
                    state <= S1;
                    Counter <= 1;
                end if;
            when S1 =>
                Outp <= '1';
                if (Counter=40) then
                    state <= S2;
                    Counter <= 1;
                end if;
            when S2 =>
                Outp <= '0';
                if (Counter=50) then
                    state <= S3;
                    Counter <= 1;
                end if;
            when S3 =>
                Outp <= '1';
                if (Counter=20) then
                    state <= S0;
                    Counter <= 1;
                end if;
        end case;
    end if;
end process;
```

مثال ۲: طراحی یک جمع کننده قابل برنامه ریزی (قابلیت جمع کردن  $n$  عدد متوالی)



زمانبندی سیگنالهای ورودی/خروجی در شکل زیر نمایش داده شده است. سیگنال Start به مدت یک پالس ساعت فعال است و همزمان با فعال بودن آن، تعداد داده هایی که قرار است با یکدیگر جمع شوند روی پورت ورودی Count قرار می گیرد (در مثال زیر، این مقدار برابر ۴ است) بعد از غیر فعال شدن Start، اعدادی که باید با یکدیگر جمع شوند به ترتیب روی پورت Data قرار می گیرند (در مثال زیر،

این اعداد برابر با ۱۰، ۱۱، ۱۲ و ۱۳ هستند که حاصل جمع آنها ۴۶ خواهد بود). به محض اینکه آخرین داده دریافت گردید، سیگنال end1 به مدت یک پالس فعال شود و حاصل جمع روی پورت خروجی Sum قرار گیرد. مقدار پورت sum در بقیه زمانها، صفر باشد.



```
process(Clk)
begin
    if (Clk='1' and Clk'event) then
        End1_sig <= '0';
        case state is
            when Idle =>
                if (start='1') then
                    state <= S0;
                    Sum_sig <= (others=>'0');
                    Counter <= Count;
                end if;
            when S0 =>
                if ((Counter=x"00") or (Counter=x"01")) then
                    End1_sig <= '1';
                    state <= Idle;
                end if;
                Sum_sig <= Sum_sig + (x"00"& Data);
                Counter <= Counter - 1;
            end case;
        end if;
    end process;
    End1 <= End1_sig;
    Sum <= Sum_sig when (End1_sig='1') else (others=>'0');
```

چک کردن حالت استثناء، اگر تعداد داده ها صفر باشد

شرط خاتمه

۱۶ بیتی کردن ورودی

مثال ۳: جابجایی مقادیر داخل بلاک رم (مقادیر ذخیره شده در آدرسهای زوج و فرد یک بلاک رم یک کیلو بایتی را جابجا کند)

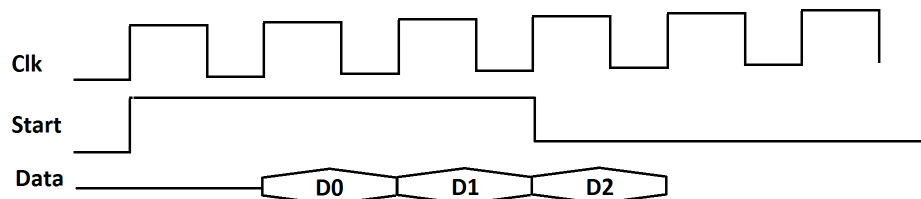
```
process(Reset, Clk)
begin
    if(Reset='1') then
        state <= S0;
        counter <= (others=>'0');
    elsif(Clk='1' and Clk'event) then
        Wr <= '0';
        case state is
            when S0 =>
                if(counter=511) then
                    state <= Stop;
                else
                    state <= S1;
                end if;
            when S1 =>
                Addr <= counter & '0';
                state <= S2;
            when S2 =>
                state <= S3;
            when S3 =>
                Temp0 <= DOut;
                Addr <= counter & '1';
                state <= S4;
            when S4 =>
                state <= S5;
            when S5 =>
                Temp1 <= DOut;
                state <= S6;
            when S6 =>
                Addr <= counter & '0';
                Din <= Temp1;
                Wr <= '1';
                state <= S7;
            when S7 =>
                Addr <= counter & '1';
                Din <= Temp0;
                Wr <= '1';
                counter <= counter + 1;
                state <= S0;
            when Stop =>
                null;
        end case;
    end if;
end process;
```

## نمونه سوال

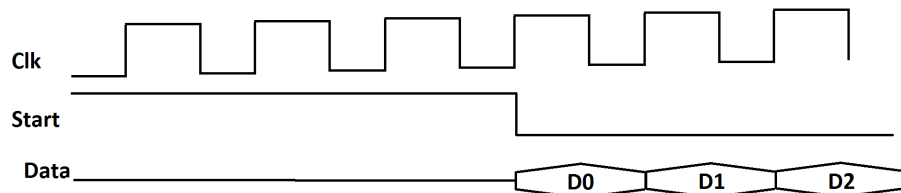
- سوال ۱: در مثال ۱، اگر مقدار اولیه counter برابر با ۲۰۰ باشد آنگاه نتیجه اجرای برنامه چیست؟
- سوال ۲: در مثال ۱، اگر فرکانس کلاک ۲۰ مگاهرتز باشد، آنگاه برای تولید همان شکل موج، چه تغییراتی باید در برنامه ایجاد کرد؟
- سوال ۳: در مثال ۲، اگر شرط ( $Counter=x"00"$ ) حذف گردد و حالت استثناء رخ دهد، خروجی برنامه چیست؟ (هنگامی که Start فعال می شود مقدار Count صفر باشد)
- سوال ۴: در مثال ۲، اگر شرط  $if((Counter=x"00")or(Counter=x"01"))then$  را به صورت زیر تغییر دهید آنگاه نتیجه اجرای برنامه چه خواهد بود؟

$if(Counter=x"00") then$

- سوال ۵: در مثال ۲، فرض کنید که سیگنال Start بیش از یک پالس فعال باشد و داده های ورودی طبق همان زمانبندی قبلی وارد شود آنگاه برنامه را به نحوی تغییر دهید که با هر بار فعال شدن Start (تغییر از صفر به یک) فقط یک بار حاصلجمع محاسبه گردد. (در برنامه کنونی اگر حاصلجمع تولید شود و به حالت Idle برگردیم و Start هنوز صفر نشده باشد، مجددا حاصلجمع محاسبه می گردد)
- تست پنج را به نحوی تغییر دهید که سیگنال Start، به مدت ۱۰ پالس فعال باشد و همان چهار عددی که در تست پنج قبلی بودند با همدیگر جمع شوند.



- سوال ۶: در مثال ۲، برنامه را به نحوی تغییر دهید که اگر سیگنال Start بیش از یک پالس فعال باشد و شروع ارسال داده ها به مارجول، وقتی باشد که Start صفر می شود. تست پنج را به نحوی تغییر دهید که سیگنال Start، به مدت ۱۰ پالس فعال باشد و همان چهار عددی که در تست پنج قبلی بودند با همدیگر جمع شوند.



- سوال ۷: برنامه ای بنویسید که محتوای یک بلاک رم ۱۰۲۴ بایتی را با همدیگر جمع بزند و حاصل جمع را در آدرس صفر حافظه بنویسد (بایت کم ارزش حاصلجمع را در آدرس صفر، بایت بعدی را در آدرس ۱ و ...)
- سوال ۸: برنامه ای بنویسید که تعداد عناصر یک حافظه بلاک رم ۱۰۲۴ بایتی را که برابر با محتویات آدرس صفر هست را شمارش کند و در آخرین بایت حافظه بنویسد. (مثلا اگر در خانه صفر حافظه عدد ۵۵ ذخیره شده باشد آنگاه تعداد دفعات تکرار عدد ۵۵ در کل حافظه را محاسبه می کند فرض کنید تعداد دفعات تکرار از ۲۵۵ بار بیشتر نمیشود)