

به نام خدا  
گزارشکار آزمایشگاه fpga  
ازمایش چهارم  
**FIFO**  
علیرضا رحمان ستایش  
ایمان استوار

## صورت مسئله:

### : First In First Out (FIFO)

در این آزمایش میخواهیم فیفو پیاده سازی کنیم  
فیفو نوعی حافظه واسط است که در سیستم ها کاربرد مختلفی دارد  
فیفو حافظه ای dual port میباشد که یک port مخصوص write و یک پرت مخصوص read  
میباشد موقع read قدیمی ترین داده هنوز خوانده نشده بر روی پرت خروجی قرار داده میشود  
فیفو دارای فلگ هایی میباشد که نشان دهنده شرایط خالی بودن پر بودن یا تقریباً خالی بودن یا تقریباً  
پر بودن را نشان میدهند  
از کاربرد های آن انتقال داده بین دو ماژول با کلاک مختلف میباشد

### طرح ( پیاده سازی )

ماژولی به نام فیفو نوشته شده است که شامل پارامتر های زیر میباشد

#parameter

FIFODATAWIDTH=8,

FIFOLENGTH=10,

COUNTERWIDTH=4,

NEAREMPTY=2,

NEARFULL=8

(

بوسیله این پارامتر ها در موقع سنتز طول فیفو، عرض فیفو، عرض شمارنده، (که متناسب با طول  
فیفو باید انتخاب شود به طوری که عددی به تعداد بیت COUNTERWIDTH بتواند  
FIFOLENGTH را در خود بدون سر ریز جای دهد) ، تعداد خانه هایی که اگر از فیفو پر شود  
NEAREMPTY حساب است و تعداد خانه هایی که اگر از فیفو پر شود NEARFULL حساب میشود

ورودی های ماژول:

```
(  
input [FIFODATAWIDTH-1:0] dataIn,  
output reg [FIFODATAWIDTH-1:0] dataOut=0,  
input push,  
input pop,  
output reg nearEmpty,  
output reg nearFull,  
output reg full,  
output reg empty,  
output reg middel  
);
```

dataIn : پرت ورودی فیفو

dataOut: پرت خروجی فیفو

Push: سیگنالی که لبه بالا رونده آن به صورت آسنکرون باعث قرار داده شدن داده از پرت خروجی به درون فیفو می شود

Pop: سیگنالی که لبه بالا رونده آن باعث قرار داده شدن قدیمی ترین داده درون روی پرت خروجی فیفو میشود

nearEmpty, nearFull, full, empty, middel : فلگ هایی که به ترتیب وقتی فیفو در میانی راه هست وقتی خالی هست وقتی پر هست و وقتی تقریباً پر هست و وقتی تقریباً خالی است یک میشوند

رجیستر های داخلی ماژول:  
حافظه اصلی ماژول:

```
reg [FIFODATAWIDTH-1:0] memBuff [FIFOLENGTH:0];
```

شمارنده ای که مقدار آن خانه ای است که قرار اسیت داده ورودی در صورت دریافت پالس write در آن نوشته شود.(اشاره گر به آن خانه حافظه):

```
reg [COUNTERWIDTH-1:0] writeCounter=0;
```

شمارنده ایست که به خانه ای از حافظه که قرار است در صورت آمدن پالس read آن خانه حافظه روی پرت خروجی قرار بگیرد اشاره میکند

```
reg [COUNTERWIDTH-1:0] readCounter=0;
```

مشخص میکند که خانه بعدی که باید توسط writeCounter به آن اشاره شود چیست با توجه به circular بودن فیفو باید nextWriteCounter به شکلی تغییر کند که وقتی شمارنده یکی یکی جلو رفت و به آخر حافظه فیفو رسید دوباره به اولین خانه فیفو برگردد

```
reg [COUNTERWIDTH-1:0] nextWriteCounter=1;
```

مشخص میکند که خانه بعدی که باید توسط readCounter به آن اشاره شود چیست با توجه به circular بودن فیفو باید nextReadCounter به شکلی تغییر کند که وقتی شمارنده یکی یکی جلو رفت و به آخر حافظه فیفو رسید دوباره به اولین خانه فیفو برگردد

```
reg [COUNTERWIDTH-1:0] nextReadCounter=1;
```

در این رجیستر فاصله بین writeCounter و writeCounter وجود دارد و از روی آن فلگ های nearEmpty, nearFull,full,empty,middel مشخص میشوند

```
reg [COUNTERWIDTH-1:0] distance ;
```

پروسه های always در ماژول: در ماژول سه پروسه always وجود دارد اولین پروسه به شرح زیر است :

در این پروسه کامبینیشنال مقادیر nextWriteCounter و nextReadCounter همواره مشخص است و مشخص میکند بعد از این writeCounter و writeCounter باید به کدام خانه حافظه اشاره کند تا این شمارنده ها به طور صحیح و circular روی فیفو حرکت کنند

همچنین در این پروسه distance که فاصله بیت writeCounter و readCounter است مشخص میشود توجه شود با توجه به circular بودن حافظه distance در دو حالتی که readCounter از writeCounter جلو تر است و در حالتی که writeCounter یک دور حافظه را دور زده و به اول فیفو رفته متفاوت است و نحوه محاسبه distance در هر حالت در کد زیر مشخص است

کار دیگر این پروسه همان طور که مشخص است تعیین فلگ ها با توجه به distance میباشد

```
always @(*) begin
```

```
    if(writeCounter==(FIFOLENGTH))
```

```
        nextWriteCounter=0;
```

```
    else
```

```
        nextWriteCounter=writeCounter+1;
```

```
    if(readCounter==(FIFOLENGTH))
```

```
        nextReadCounter=0;
```

```
    else
```

```
        nextReadCounter=readCounter+1;
```

```
    if(writeCounter>=readCounter)
```

```
        distance=writeCounter-readCounter;
```

```
    else
```

```
        distance=FIFOLENGTH-readCounter+writeCounter+1;
```

```
    if(distance==0)begin
```

```
        nearEmpty=0;
```

```
    nearFull=0;

    full=0;

    empty=1;

    middel=0;

end

else if(distance<=NEAREMPTY)begin

    nearEmpty=1;

    nearFull=0;

    full=0;

    empty=0;

    middel=0;

end

else if(distance==FIFOLENGTH)begin

    nearEmpty=0;

    nearFull=0;

    full=1;

    empty=0;

    middel=0;

end

else if(distance>=NEARFULL)begin

    nearEmpty=0;

    nearFull=1;

    full=0;
```

```

        empty=0;
        middel=0;
    end
    else begin
        nearEmpty=0;
        nearFull=0;
        full=0;
        empty=0;
        middel=1;
    end
end
end

```

دو پروسه `always` دیگر به شرح زیر است:

در این پروسه انجام عمل نوشتن بر روی حافظه صورت میگیرد البته این عمل تا قبل از پر شدن فیفو (در واقع رسیدن `nextWriteCounter` به `readCounter`) صورت میگیرد و بعد از پر شدن فیفو دیگر نمیتوان داده ای روی آن نوشت

```

always @ (posedge push ) begin
    if(nextWriteCounter!=readCounter) begin
        writeCounter<=nextWriteCounter;
        memBuff[writeCounter]<=dataIn;
    end
end
end

```

در این پروسه انجام عمل خواندن از روی حافظه صورت میگیرد البته این عمل تا قبل از خالی شدن فیفو (در واقع رسیدن writeCounter به readCounter) صورت میگیرد و بعد از خالی شدن فیفو دیگر نمیتوان داده از روی آن خواند

```
always @ (posedge pop ) begin
    if(readCounter!=writeCounter) begin
        readCounter<=nextReadCounter;

        dataOut<=memBuff[readCounter];

    end

end

end
```

تا اینجا ماژول فیفو را شرح دادیم به شرح تاپ ماژول میپردازیم

درون topmodule ماژول های my\_ FIFO debouncer2 debouncer1 BCDToSevenSegment2 BCDToSevenSegment1 ورودی تاپ ماژول از دو سویچ روی برد یکی به عنوان تامین کنند پالس push و یکی pop استفاده شده که بعد از دیبانس شدن به ماژول فیفو داده میشود

پرت وردی فیفو به دیپ سویچ ها وصل است و از انها ورودی میگیرد همچنین پرت ورودی به یکی از ماژول های BCDToSevenSegment2 رفته و روی یکی از سون سگمنت ها نمایش داده میشود تا ورودی فیفو روی ان دیده شود

پرت خروجی فیفو به ماژول BCDToSevenSegment1 میرود و از انجا به سون سگمنت میرود تا خروجی فیفو نمایش داده شود



فلگ های فیفو نیز به LED ها وصل است تا STATUS فیفو روی ال ای دی ها نمایش داده شود.

### **نتیجه:**

بعد از انجام شبیه سازی کد را روی برد انتقال دادیم و نتیجه مطلوبی که از فیفو انتظار میرفت را مشاهده نمودیم