چکیده:

این ازمایش درباره ی پردازش تصویر ساده با استفاده از HLS است که ما در این جا از Catapult استفاده کرده ایم.

مقدمه:

در این ازمایش ابتدا عکسی را به وسیله ی matlab به پیکسل های ان تبدیل کرده سپس در کرنل مخصوصی ضرب می شوند و توسط catapult اجرا می شود و 1 ram و یک rom می نویسیم و دیتا ی جدید را ذخیره می کنیم و سپس به وسیله ی متلب ان را نمایش می دهیم

انجام پروژه:

مرحله1:

ابتده عکس داده شده را به پیکسل هایش تبدیل می کنیم که چون عکس سیاه سفید است از هر پیکسل فقط 1 مقدار داریم. این کار را با متلب و دستور imread انجام می دهیم

مرحله2:

سپس کرنل خود را پیدا کردیم(3 × 3 Gaussian Blur)

مرحله3:

برنامه ای به زبان C می نویسیم و الگریتم ضرب را در ان همان طور که گفته شده پیاده می کنیم

```
void Convolotion(ac_fixed<8,8,false,AC_TRN,AC_SAT> in[262144] , ac_fixed<8,8,false,AC_TRN,AC_SAT> out[262144])
{
    ac_fixed<8,0,false,AC_TRN> kernel[9];
    kernel[0] = 0.0625;
    kernel[1] = 0.125;
    kernel[2] = 0.0625;
    kernel[3] = 0.125;
    kernel[4] = 0.025;
    kernel[5] = 0.125;
    kernel[6] = 0.0625;
    kernel[7] = 0.125;
    kernel[8] = 0.0625;

    ac_fixed<8,8,false,AC_TRN,AC_SAT> Result;

    ac_int<11,true> Drow;
    ac_int<11,true> Dcolumn;
    ac_int<4,true> Krow;
    ac_int<4,true> Kcolumn;

    ac_int<11,true> conv1;
    ac_int<11,true> conv2;

    ac_int<21,false> Data_Select;
    ac_int<5,false> Kernel_Select;

    for(Dcolumn=0;Dcolumn<512;Dcolumn++)
    {
        for(Drow=0;Drow<512;Drow++)
        {
            Result=0;
            for(Kcolumn=0;Kcolumn<3;Kcolumn++)
            {
                for(Krow=0;Krow<3;Krow++)
                {
                    conv1 = (Dcolumn-(1-Kcolumn));
                    conv2 = (Drow-(1-Krow));

                    if(conv1<0 || conv1>511 || conv2<0 || conv2>511)
                        Result = Result + 0;
                    else
                    {
                        Data_Select = (conv2 * 512) + (conv1);
                        Kernel_Select = ((Krow*3) + Kcolumn);

                        Result = Result + (in[Data_Select] * kernel[Kernel_Select]);
                    }
                }
            }
            Data_Select = (Drow * 512) + Dcolumn;
            out[Data_Select] = Result;
        }
    }
}
```

کد در فایل جدا نیر قرار داده شده است.

مرحله 4:

ابتدا با استفاده از catapult کد را به Verilog تبدیل می کنیم و جواب را به modelsim می بریم

سپس 1 ماژول ram و یک rom به طور جدا با استفاده از ادرس و سیگنال کنترلی تولید شده توسط catapult می نویسیم و با نوشتن تست بنچ جدا و وصل کردن 3 ماژول ان را اجرا می کنیم و دیتا را إخیره می کنیم

```
module Ram(Address,Data_In,Clock,Rst,WriteEnable,finish);

  input[17:0] Address;
  input Clock,Rst,WriteEnable;
  input [7:0] Data_In;
  input finish;

  reg[7:0] Ram[0:262144];

  always@(posedge Clock) begin
    if(finish)begin
      $writememh("result.txt",Ram);
    end

  end

  always@(posedge WriteEnable)begin
  if(!Rst) begin
      Ram[Address] <= Data_In;
    end
  end


endmodule
```

```
module Rom(Address,Data_out,Clock,Rst);

  input[17:0] Address;
  input Clock,Rst;
  output reg[7:0] Data_out;

  reg[7:0] Rom[0:262144];

  always@(posedge Clock) begin
   if(Rst) begin
      $readmemb("Img.txt",Rom);
  end

    Data_out <= Rom[Address];
  end

endmodule
```

```
Convolotion U1(
  Clock, Rst, in_rsc_singleport_data_in, in_rsc_singleport_addr, in_rsc_singleport_re,
     in_rsc_singleport_we, in_rsc_singleport_data_out, out_rsc_singleport_data_in,
     out_rsc_singleport_addr, out_rsc_singleport_re, out_rsc_singleport_we, out_rsc_singleport_data_out
);

Ram U2  (out_rsc_singleport_addr,
         out_rsc_singleport_data_in,
         Clock,
         Rst,
         out_rsc_singleport_we,
         finish);

Rom U3 (in_rsc_singleport_addr,
        in_rsc_singleport_data_out,
        Clock,
        Rst);
endmodule
```

```
`timescale 1ps/1ps

module Test();

  reg Clock,Rst,finish;

  Top_File UT(Clock,Rst,finish);


  always #6 Clock=~Clock;

  initial begin
  finish=0;
  Clock=0;
  Rst=1;
  #18
  Rst=0;
  #69300000
  finish=1;
  #50
  finish=0;
  #150

  $stop;


  end

endmodule
```

کد در فایل modelsim قرار داده شده است.

مرحله5:

جواب ذخیره شده را به متلب برده و ان را با imshow نمایش می دهیم و عکس نهایی را می بینیم می توانیم با imwrite ان را ذخیره کنیم

a:input *b:output*

می بینیم که عکس همان طور که انتظار می رفت Blur شده است

مرحله۶:

قبل از loop pipelining :

Latency Cycle: 5767165, Latency Time: 115343300.00, Slack: 10.57

Throughput Cycle: 5767681, Throughput Time: 115353620.00, Area: 505.82

:Main Loop Pipeline

Latency Cycle: 4718790, Latency Time: 94371800.00, Slack: 10.88

Throughput Cycle: 4718592, Throughput Time: 94371840.00, Area: 667.21

:First Loop Pipeline (Row of pixels)

Latency Cycle: 4718790, Latency Time: 94371800.00, Slack: 9.75

Throughput Cycle: 4718594, Throughput Time: 94371880.00, Area: 614.12

:Second Loop Pipeline (Column of pixels)

Latency Cycle: 4719101, Latency Time: 94382020.00, Slack: 10.88

Throughput Cycle: 4719617, Throughput Time: 94392340.00, Area: 573.95

:Third Loop Pipeline (Row of kernel)

Latency Cycle: 4980733, Latency Time: 99614660.00, Slack: 10.88

Throughput Cycle: 4981249, Throughput Time: 99624980.00, Area: 521.43

:Fourth Loop Pipeline (Column of kernel)

Latency Cycle: 5767165, Latency Time: 115343300.0.00, Slack: 10.566727

Throughput Cycle: 5767681, Throughput Time: 115353620.0.00, Area: 505.8216669600001

تمام حالات بالا برای II=2 انجام شده است

مرحله ۷:

قبل از loop Unrolling :

Latency Cycle: 5767165, Latency Time: 115343300.00, Slack: 10.57

Throughput Cycle: 5767681, Throughput Time: 115353620.00, Area: 505.82

:First Loop Unroll (Row of pixels)

Latency Cycle: 6041087, Latency Time: 120821740.00, Slack: 10.566727

Throughput Cycle: 6041602, Throughput Time: 120832040.00, Area: 560.4563520700001

:Second Loop Unroll (Column of pixels)

Latency Cycle: 6041085, Latency Time: 120821700.00, Slack: 10.566727

Throughput Cycle: 6041601, Throughput Time: 120832020.00, Area: 553.47773927

:Third Loop Unroll (Row of kernel)

Latency Cycle: 4980733, Latency Time: 99614660.00, Slack: 9.433713

Throughput Cycle: 4981249, Throughput Time: 99624980.0, Area: 542.64190249

:Fourth Loop Unroll (Column of kernel)

Latency Cycle: 3407869, Latency Time: 68157380.0, Slack: 5.8345590000000005

Throughput Cycle: 3408385, Throughput Time: 68167700.0, Area: 620.6673237

تمام کار های بالا با partial unrollبا ii=3 انجام شده است (نمی توانست بدون partial انرول کند)

مرحله ۸:

با تغییر در ساختار کد می توان Area یای کمتری استفاده کرد (می توان ریسورس ها را اپتیمایز کرد)

و اگر کد به صورت Class نوشته شود با کمترین تغییر تو کد میشه ازش استفاده کرد.