代码示例

插入代码的操作主要是由 ‘lstlisting‘宏包实现的。在这段时间内，我尝试输入了不同样式的代码文件，包括了 python（计算机视觉，机器学习等课程）,verilog（计算机组成原理）,cpp（OS）,matlab（数模）等，不同代码语言内容有相似之处，但是使用方法还是略有不同, 下面进行逐一介绍。

首先鉴于代码中的注释文件可能含有中文，此时需要全局设置字符编码为 ‘utf-8‘。在文档的开头进行全局样式设置。这些样式设置是全局的，也就是说，无论你后续使用的是何种设置，这一全局的结构都会进行覆盖。

首先展示一部分在论文中插入代码的方法：

1 Matlab

行内代码插入：

```
    function GAdsa
    %% 这个部分放点代码
```

以文件形式插入：

```
function [r2,rmse] = rsquare(y,f,varargin)
% Compute coefficient of determination of data fit
    model and RMSE
%
% [r2 rmse] = rsquare(y,f)
% [r2 rmse] = rsquare(y,f,c)
%
% RSQUARE computes the coefficient of determination (R
    -square) value from
% actual data Y and model data F. The code uses a
    general version of
% R-square, based on comparing the variability of the
    estimation errors
% with the variability of the original values. RSQUARE
     also outputs the
```

```
11 % root mean squared error (RMSE) for the user's
       convenience.
12 %
13 % Note: RSQUARE ignores comparisons involving NaN
       values.
14 %
15 % INPUTS
16 %   Y        : Actual data
17 %   F        : Model fit
18 %
19 % OPTION
20 %   C        : Constant term in model
21 %              R-square may be a questionable measure
       of fit when no
22 %              constant term is included in the model.
23 %   [DEFAULT] TRUE : Use traditional R-square
       computation
24 %             FALSE : Uses alternate R-square
       computation for model
25 %                     without constant term [R2 = 1 -
       NORM(Y-F)/NORM(Y)]
26 %
27 % OUTPUT
28 %   R2       : Coefficient of determination
29 %   RMSE     : Root mean squared error
30 %
31 % EXAMPLE
32 %   x = 0:0.1:10;
33 %   y = 2.*x + 1 + randn(size(x));
34 %   p = polyfit(x,y,1);
35 %   f = polyval(p,x);
36 %   [r2 rmse] = rsquare(y,f);
37 %   figure; plot(x,y,'b-');
```

```
38 %   hold on; plot(x,f,'r-');
39 %   title(strcat(['R2 = ' num2str(r2) '; RMSE = '
       num2str(rmse)]))
40 %
41 % Jered R Wells
42 % 11/17/11
43 % jered [dot] wells [at] duke [dot] edu
44 %
45 % v1.2 (02/14/2012)
46 %
47 % Thanks to John D'Errico for useful comments and
       insight which has helped
48 % to improve this code. His code POLYFITN was
       consulted in the inclusion of
49 % the C-option (REF. File ID: #34765).
50
51 if isempty(varargin); c = true;
52 elseif length(varargin)>1; error 'Too many input
       arguments';
53 elseif ~islogical(varargin{1}); error 'C must be
       logical (TRUE||FALSE)'
54 else c = varargin{1};
55 end
56
57 % Compare inputs
58 if ~all(size(y)==size(f)); error 'Y and F must be the
       same size'; end
59
60 % Check for NaN
61 tmp = ~or(isnan(y),isnan(f));
62 y = y(tmp);
63 f = f(tmp);
64
```

```
if c; r2 = max(0,1 - sum((y(:)-f(:)).^2)/sum((y(:)-
    mean(y(:))).^2));
else r2 = 1 - sum((y(:)-f(:)).^2)/sum((y(:)).^2);
    if r2<0
    % http://web.maths.unsw.edu.au/~adelle/Garvan/
        Assays/GoodnessOfFit.html
        warning('Consider adding a constant term to
            your model') %#ok<WNTAG>
        r2 = 0;
    end
end

rmse = sqrt(mean((y(:) - f(:)).^2));
```

2 Python

行内代码插入：

```
    from PIL import Image
    from numpy import asarray
```

以文件形式插入：

```
# This is a sample Python script.

# Press Shift+F10 to execute it or replace it with your
    code.
# Press Double Shift to search everywhere for classes,
    files, tool windows, actions, and settings.


def print_hi(name):
    # Use a breakpoint in the code line below to debug
        your script.
    print(f'Hi, {name}')  # Press Ctrl+F8 to toggle the
        breakpoint.


# Press the green button in the gutter to run the script.
if __name__ == '__main__':
    print_hi('PyCharm')

```

```
# See PyCharm help at https://www.jetbrains.com/help/pycharm/
```

3 Verilog

行内代码插入：

```
module trafic_light(
    input switch0,switch1,
    output R,G,Y
);
 assign Y =  switch1;
 assign R =  switch0 & ~ switch1;
 assign G = ~switch0 & ~ switch1;
endmodule
```

文件插入：

```
module INST_ROM(

 input  [31:0] address,
 output [31:0] inst
 );
 //只读的指令存储器，输入为地址，输出为指令，在computer中使用
   wire [31:0] ram [0:31];
 assign ram[5'h00]=32'h00002820; //Add $a1, $0, $0   ;
 assign ram[5'h01]=32'h8CB10000; //Lw   $s1, 0($a1)  ; ram0 = 30
 assign ram[5'h02]=32'h8CB20004; //Lw   $s2, 4($a1)  ; ram1 = 20
 assign ram[5'h03]=32'h3e330001; //Lui $s3, $s1, 16'h0001 高位置数;
 assign ram[5'h04]=32'h02519824; //And $s3, $s1, $s2 与，结果为20;
 assign ram[5'h05]=32'h02519825; //Or  $s3, $s1, $s2 或，结果为30;
 assign ram[5'h06]=32'h02519826; //Xor $s3, $s1, $s2 异或，结果为10;

 assign ram[5'h07]=32'h02328820; //Add $s1, $s1, $s2 ;
 assign ram[5'h08]=32'h8CB20008; //Lw   $s2, 8($a1)  ; ram2 = -10
 assign ram[5'h09]=32'h12320001; //Beq $s1, $s2, 1   ; 不会跳转
 assign ram[5'h0A]=32'hACA0000C; //Sw   $0, 12($a1)  ; ram[3] = 40 如果跳转，则后续读取是原本的数值40，没有跳转数据应该是0
 assign ram[5'h0B]=32'h8CB1000C; //lw   $s1, 12($a1) ;
 //此时ram3为0
 //再次加载
 assign ram[5'h0C]=32'h8CB10000; //Lw   $s1, 0($a1)  ; ram0 = 30
 assign ram[5'h0D]=32'h8CB20004; //Lw   $s2, 4($a1)  ; ram1 = 20
 assign ram[5'h0E]=32'h02328822; //Sub $s1, $s1, $s2 ; 相减
 assign ram[5'h0F]=32'hAE40000C; //Sw   $2, 12($a1)  ; ram[3] = 20
 assign ram[5'h10]=32'h8CB20008; //Lw   $s2, 8($a1)  ; ram2 = -10
 assign ram[5'h11]=32'h12320001; //Beq $s1, $s2, 1   ; 跳转
 assign ram[5'h12]=32'hACA0000C; //Sw   $0, 12($a1)  ;  如果跳转，则后续读取是原本的数值20，没有跳转数据应该是0
 assign ram[5'h13]=32'h8CB1000C; //lw   $s1, 12($a1) ;
 assign ram[5'h14]=32'h8CB1000C; //lw   $s1, 12($a1) ;
```

```
    assign inst=ram[address[6:2]];
 endmodule

\end{lstlisting}
\subsection{Data MEM模块设计与实现}
为测试lw和sw指令，数据存储器件也存有一些数据，其编码如下：
\begin{lstlisting}[style={prettyverilog}]
  module DATA_RAM(
        input         Clock,
        output[31:0] dataout,
        input [31:0] datain,
        input [31:0] addr,
        input         write , read
    );
//RAM 用于存储数据，相当于内存
   reg [31:0] ram [0:31];

    assign dataout = read ? ram[addr[6:2]] : 32'hxxxxxxxx;
// 时钟下降沿的时候写内存
    always @ (posedge Clock) begin
            if (write) ram[addr[6:2]] = datain;
    end

    integer i;

    initial begin
    //set data_ram number
            ram[0]= 30;
            ram[1]= 20;
            ram[2]= 10;
            ram[3]= 40;
    end

endmodule
```