PLT Project

SIP (Simplified Image Processing)
A Language for image processing
Why SIP ??

● Effectively an image processing language.
● Concept can be extended for videos
● Features included to make operations on images short and effective
● Attributes of images : Pixels

Images – basically a 2D array of pixels.
● Action on images implies action on each pixel.
What SIP can do?

- Basic data types – int, float, string, bool, pixels and images.
- Pixel – A four element tuple.
- Basic calculations: Boolean operations, Arithmetic operations, string operations, Pixel operations.
- Basic control flow: if statements, do.. while statements, while .. statements, break.. continue statements, for loops.
Operators

- Arithmetic operators
  - '+' operator to add floating point numbers, integers and pixels.
    
a + b returns the sum of a and b// or individual color components for a pixel
  - '-' operator used to subtract integers, floating point numbers or pixels.
    
a - b returns the difference of a and b/ or individual color components for a pixel
  - '/' operator to divide floating point numbers and integers
    
a / b returns the quotient of a and b.
  - '*' operator to multiply integers and floating point numbers.
    
a * b returns the product of a and b.
Operators

- '%' operator to return the remainder (or the modulus operator). Here, \( a \% b \) returns the remainder obtained when \( a \) is divided by \( b \).

- Boolean operators:
  - \( A + B \) returns the result of the logical OR
  - \( A \times B \) returns the result of the logical AND
  - \( ! / - A \) will return the complement of \( A \)
Usage

- Environment:
  - Ocaml
  - gcc
  - cImg (used only for displaying images)

- Steps
  - make clean
  - make
  - make input.svip
Sample Code for cropping

```c
string s = input("Enter the path to the image file: ");
display("The path you entered is: ", s);
image i1 = open(s);

image crop(image i, int a, int b)
{
    image df[a][b];
    for(int k = 0; k < b; k++)
    for(int j = 0; j < a; j++)
        df[k][j] = i[k][j];
    return df;
}

s = input("Enter the path for the output image file: ");
display("The path you entered is: ", s);
save(crop(i1, 100, 100), s);
input();
```
Tutorial for SIP – Demo 1 Results

Cropping
Sample code for Image edge detection

- string s = input("Enter the path to the image file: ");
- display("The path you entered is: ",s);

- int a[9] = [-1,-1,-1, -1,8,-1, -1,-1,-1];
- image from = open(s);
- image to = open(s);

- int pos;
- int sum1,sum2,sum3;
- for(int k = 0;k < from.height - 3; k+=1)
  {
    for(int l = 0; l< from.width - 3; l+=1)
      {
        sum1 = sum2 = sum3 = 0;
        for( int i = 0; i < 3; i+=1) {
          for(int j = 0; j < 3; j+=1)
            pos = (3 * i) + j;  sum1 += from[(i + k)][(j+l)].C1 * a[pos]; sum2 += from[(i + k)][(j+l)].C2 * a[pos]; 
            sum3 += from[(i + k)][(j+l)].C3 * a[pos];
        }
      }
  }

  to[(k + 4)][(l+4)].C1 = sum1;
Tutorial for SIP – Demo 2 Results

Edge detection
Project Architecture

SIP source code → Scanner → Parser → 

Semantic analysis → C++ code generation → 

Intermediate C++ code → C++ compiler → 

Executable file
Summary

- First and foremost – **DO NOT** use Windows for compiler development ...
- Design early!
- Get everyone involved early.
- Digital VLSI doesn't go well with PLT.