

α Project



Dr. Fabian Vargas

Darcio Prestes

Diogo Brum

Djones Lettnin

Gustavo Rodrigues

Letícia Bolzani



Outline

- Introduction;
- HW Description;
- Study Case;
 - Serial Protocol;
 - Original Algorithm;
 - FT Algorithm;
 - BIST;
- Goals



Introduction

- Fault Tolerance in Embedded Systems under Electromagnetic Interference:
 - Electromagnetic Compatibility (EMC);
 - FT Algorithms;
 - Built-In Self Test (BIST);
 - Test and Control Sets:
 - Test Set : harden system;
 - Control Set : original system;



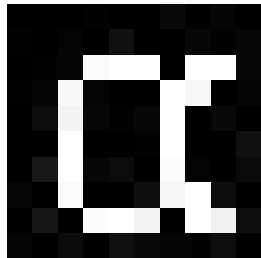
HW Description (MSP430F149)

- Basic Features of :
 - Memory - 2KB (RAM), 60KB (FLASH)
 - 8MHz (Clock);
 - RISC
 - Ultra Low Power
- Assembly and C-ANSI Programming;

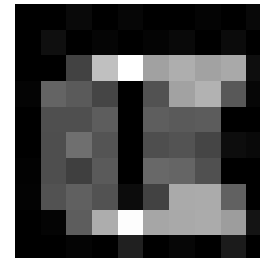
Study Case

- Blur Filter applied in simple image;
- Most ALU's operations used:
 - sum and subtraction;
 - multiply and division;

Original image



Final image





Serial Protocol

- Structure blocks

| start | instruction | data | final |

- start: source and destiny address;
- instruction: comand;
- data: data size and data block;
- final: checksum;



Original Algorithm

```
void Blur(unsigned char *imageIn, int width, int high) {
    int r, c;
    unsigned char imag[width*high];
    for (r=width; r<(width*(high-1)); r+=width)
        for (c=1; c < (width-1); c++)
            {
                imag[c+r] = (
                    imageIn[r-1+c] + imageIn[r+c]
                    + imageIn[r+c+1]) /3;
            }

    for (r=width; r<(width*(high-1));r++)
        {
            imageIn[r] = imag[r];
        }
}
```



FT Algorithm

```
void BlurFT(unsigned char *imageIn0, unsigned char *imageIn1, int width0,
            int width1, int high0 , int high1)
{
    int r0,r1,c0,c1,i0,i1;
    unsigned char imag0[width0 * high0];
    unsigned char imag1[width1 * high1];
    ecf = 10;
    if ((width0 != width1) || (high0 != high1))
    {
        Erro(1); // Erro na comparacao dos parametros
        return;
    }
    [...]
}
```


FT Algorithm(2)

```
[...]
for (r0=width0,r1=width1; r0<(width0*(high0-1)) && r1<(width1*(high1-1));
    r0+=width0,r1+=width1)
{
    if ((width0 != width1) || (high0 != high1))
    {
        Erro(1); // Erro na comparacao dos parametros
        return;
    }
    for (c0=1,c1=1 ; c0 < (width0-1) && c1< (width1-1); c0++, c1++)
    {
        if ((width0 != width1))
        {
            Erro(1); // Erro na comparacao dos parametros
            return;
        }
        imag0[c0+r0] = (imageIn0[r0-1+c0] + imageIn0[r0+c0]+ imageIn0[r0+c0+1]) /3;

        imag1[c1+r1] = (imageIn1[r1-1+c1] + imageIn1[r1+c1]+ imageIn1[r1+c1+1]) /3;

        if ((r0 != r1) || (c0 != c1) || (imageIn0[r0-1+c0] != imageIn1[r1-1+c1])
            || (imageIn0[r0+c0] != imageIn1[r1+c1])
            || (imageIn0[r0+c0+1] != imageIn1[r1+c1+1]))
        {
            Erro(1); // Erro na comparacao dos parametros
            return;
        }
    }
}
[...]
```

FT Algorithm(3)

```
[...]
for (r0=width0,r1=width1; r0<(width0*(high0-1)) && r1<(width1*(high1-1));
    r0+=width0,r1+=width1)
{
    if ((width0 != width1) || (high0 != high1))
    {
        Erro(1); // Erro na comparacao dos parametros
        return;
    }
    for (c0=1,c1=1 ; c0 < (width0-1) && c1< (width1-1); c0++, c1++)
    {
        if ((width0 != width1) || (high0 != high1))
        {
            Erro(1); // Erro na compara#1"7o das imagens
            return;
        }
        imageIn0[r0+c0] = imag0[r0+r0];
        imageIn1[r1+c1] = imag1[r1+c1];

        if ((c0!=c1) || (r0 != r1) || (imag0[r0+c0] != imag1[r0+c0]) ||
            (imag0[r1+c1] != imag1[r1+c1]))
        {
            Erro(2); // Erro na compara#1"7o das imagens
            return;
        }
    }
}
if (ecf != 10)
{
    Erro(3); // Erro no controle ecf1
    return;
}
ecf=20;
}
```

Original size code: 439 bytes
FT size code: 1720 bytes

Transparent BIST

Transparent BIST - is used test the memory RAM from MSP430.

Address (Hex.)	Function	Access	
0FFFh	Interrupt Vector Table	ROM	Word/Byte
0FFE0h			
0FFDFh	Program Memory Branch Control Tables Data Tables...	ROM	Word/Byte
0200h	Data Memory	RAM	Word/Byte
01FFh	16-Bit Peripheral Modules	Timer, ADC, ...	Word
0100h			
0FFh	8-Bit Peripheral Modules	I/O, LCD 8bT/C, ...	Byte
010h	Special Function Registers	SFR	Byte
0Fh			
0h			

Size: 2 Kbytes



Transparent BIST implementation

1^a Step: algorithm was implemented in C.

Problem: this method use alocation of variables on RAM.

2^a Step: algorithm will be implemented in assembly.

Solution: this method only use register from MSP430.



Goals

- Tests with Electromagnetic Interference;
- Tests with BIST;
- Optic Fiber for serial communication;