

Sprawozdanie

Obliczenia
sumy kontrolnej
CRC
w systemach
czasu rzeczywistego

Piotr Heinzelman
Wydział Elektryczny

1. Program

Program napisany w assemblerze, skompilowany dla systemów operacyjnych Linux, dla procesorów rodziny AMD.

główny podprogram używa wyliczonych wcześniej wartości by bez zbędnej zwłoki podać wynik obliczenia sumy kontrolnej. Takie rozwiązanie powoduje większe zużycie pamięci operacyjnej (większe o 2x256 bajtów więc pomijalnie mało).

Na obliczenie sumy dla pojedynczego znaku wykonywane są:

- 3 operacje odczytania bajtu z pamięci
(największy narzut czasowy kilkanaście cykli zegara na jedną operację)
- kilka operacji MOV i XOR na rejestrach (wykonywane w ALU trwają 1 do kilku cykli).

nie wykonujemy żadnych trudnych operacji matematycznych typu mnożenie i dzielenie, mamy kilka dodawań (+1).

program odczytuje dane z pliku data.txt

w formacie hex rozdzielonym spacją, wielkość liter nie jest istotna.

30 40 50 60 70 80 90 A1 BB C0 D9

liczbę powtórzeń podajemy po nazwie programu w czasie uruchamiania:

./hello 10

1 cykl

dane: 506f6c69746563686e696b61

słowo: "Politechnika"

suma kontrolna: 12A3

zgodna z programem CRC16

czasy uzyskane wyniki dla leciwego Thinkpad x230

1.000.000 cykli - 0 sek.

10.mln - 1 sek.

100.mln - 3 sek.

1000.mln - 34 sek.

10.000.mln - 335 sek.

1 cykl - 0.0000000335 sek

1 cykl - 0.0000335 msek

1 cykl - 0.0335 usek

oneCRC:

```
    mov R9B, byte [rbx] ; dh = *pMessage
    inc rbx             ; Message++
    mov dl, ah          ; index = HiByte ^ *pMessage
    xor dl, R9B         ; index = dl ; Index = HiByte ^ *pMessage++;

    mov ah, byte [ myHigh + rdx ]
    xor ah, al          ; HiByte = LoByte ^ aCRCHi[Index];
    mov al, byte [ myLow + rdx ]
                    ; LoByte = aCRCLo[Index];
    dec ch
    jnz oneCRC
```