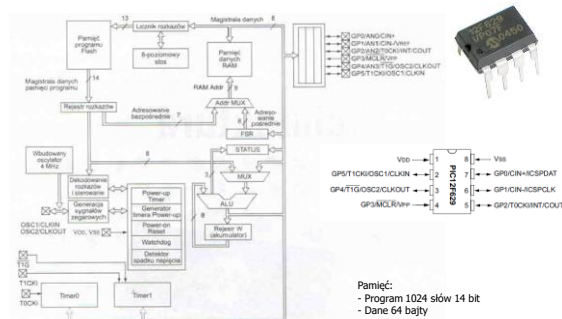


Wykład 5

- Rekurencja w programowaniu

1

Architektura PIC 12F629



2

Lista rozkazów

[illegible]

gdzie:

- f – 7-bitowy adres rejestru (0...127)
- F – zawartość rejestru o adresie f (0...255)
- W – zawartość rejestru roboczego (0...255)
- d – adres wyniku operacji (0,1);
- gdy $d = 0$, wynik operacji jest przysyłany do W
- gdy $d = 1$, wynik operacji jest przysyłany do F
- b – numer bitu (0...7)
- k – 8-bitowa stała liczbowa (0...255)
- s – 11-bitowa etykieta (0...2047)

3

Przykładowy program

```
#include "12f629.h"
```

```
int nwd(int a, int b) {
    while (a!=b) {
        if (a>b) a=a-b;
        else b=b-a;
    }
    return a;
}
```

```
void main() {
    int x,y,z;

    x=24;
    y=30;
    z=nwd(x,y);
}
```

4

Kompilacja

```

Carry EQU 0
Zero EQU 2
a EQU 0x23
b EQU 0x24
x EQU 0x20
y EQU 0x21
z EQU 0x22

;int mod(m)

mwd ; while (a)

m001 MOVF a,W ; if (a>b)
      XORWF b,W
      BTFSC 0x3,Zero_
      GOTO m001

      MOVF a,W
      SUBWF b,W
      MOVF a,W
      XORWF b,W
      BTFSC 0x3,Carry
      XORLW 128
      ANDLW 128
      BTFSS 0x3,Zero_
      GOTO m002

      MOVF b,W
      SUBWF a,W
      GOTO m001

m002 MOVF a,W ; else b=
      SUBWF b,W ;

      GOTO m001 ; }

m003 MOVF a,W ; return a;
      RETURN ;}

```

[illegible]

5

Funkcja rekurencyjna

$$n! = 1 * 2 * 3 * \dots * n$$
$$0! = 1$$

$$n! = \begin{cases} 1 & \text{gdý } n < 2 \\ (n-1)! * n & \text{gdý } n \geq 2 \end{cases}$$

```
def silnia(n):
    s = 1
    while n>1:
        s = s*
        n = n-
    return s
# end def
```

```
def silnia(n):
    if n<2:
        return 1
    else:
        return silnia(n-1)*n
# end def
```

6

Ciąg Fibonacciego

$$F(n) = \begin{cases} 1 & \text{dla } n < 3 \\ F(n-1) + F(n-2) & \text{dla } n \geq 3 \end{cases}$$



1	1	2	3	5	8	13	21				
---	---	---	---	---	---	----	----	--	--	--	--

```
def fib(n):
    a=b=1
    for i in range(1,n):
        c=a+b
        a=b
        b=c
    return a
# end def
```

```
def fib(n):
    if n<3:
        return 1
    else:
        return fib(n-1)+fib(n-2)
# end def
```

7

Wizualizacja rekurencji

```
def fib(n):
    print("start",n)

    if n<3: result = 1
    else: result = fib(n-1)+fib(n-2)

    print("stop",n)
    return result
# end def

w = fib(5);
print('w=', w)
```

```
start 5
start 4
start 3
start 2
stop 2
start 1
stop 1
stop 3
start 2
stop 2
start 4
stop 4
start 3
start 2
stop 2
start 1
stop 1
stop 3
stop 5
w=5
```

8

Rekurencja z tablicowaniem

```
N = 1000
mem = [0 for _ in range(N)]

def fib(n):
    global mem

    if n<3:
        return 1

    if mem[n]==0:
        mem[n] = fib(n-1)+fib(n-2)

    return mem[n]
```

9

Rekurencja z tablicowaniem

```
from functools import cache

@cache
def fib(n):
    if n<3:
        return 1
    else:
        return fib(n-1)+fib(n-2)
# end def

for i in range(50):
    print(i,fib(i))
```

10

Zamiana na postać dwójkową

```
def bin(n):
    t = []
    while n>0:
        t.append(n%2)
        n = n//2
    # end while
    t.reverse()
    for x in t:
        print(x, end='')
    # end for
# end def
```

```
def bin(n):
    if n>0:
        bin(n//2)
        print(n%2, end='')
    # end if
# end def
```

11

Zamiana na postać dwójkową

```
def bin(n):
    if n>0:
        bin(n//2)
        print(n%2, end="")
    # end if
# end def
```

```
x = int(input())
bin( x )
```

12

Pytania i zadania

1. Dana jest funkcja określona rekurecyjnie

$f(0,b) = b+1$

$f(a,0) = f(a-1, 1) \qquad a>0$

$f(a,b) = f(a-1, f(a,b-1)) \qquad a>0, b>0$

Proszę uzupełnić wartości funkcji w tabeli:

a \ b	0	1	2	3	4	5
0						
1						
2						
3						
4						
5						

13