

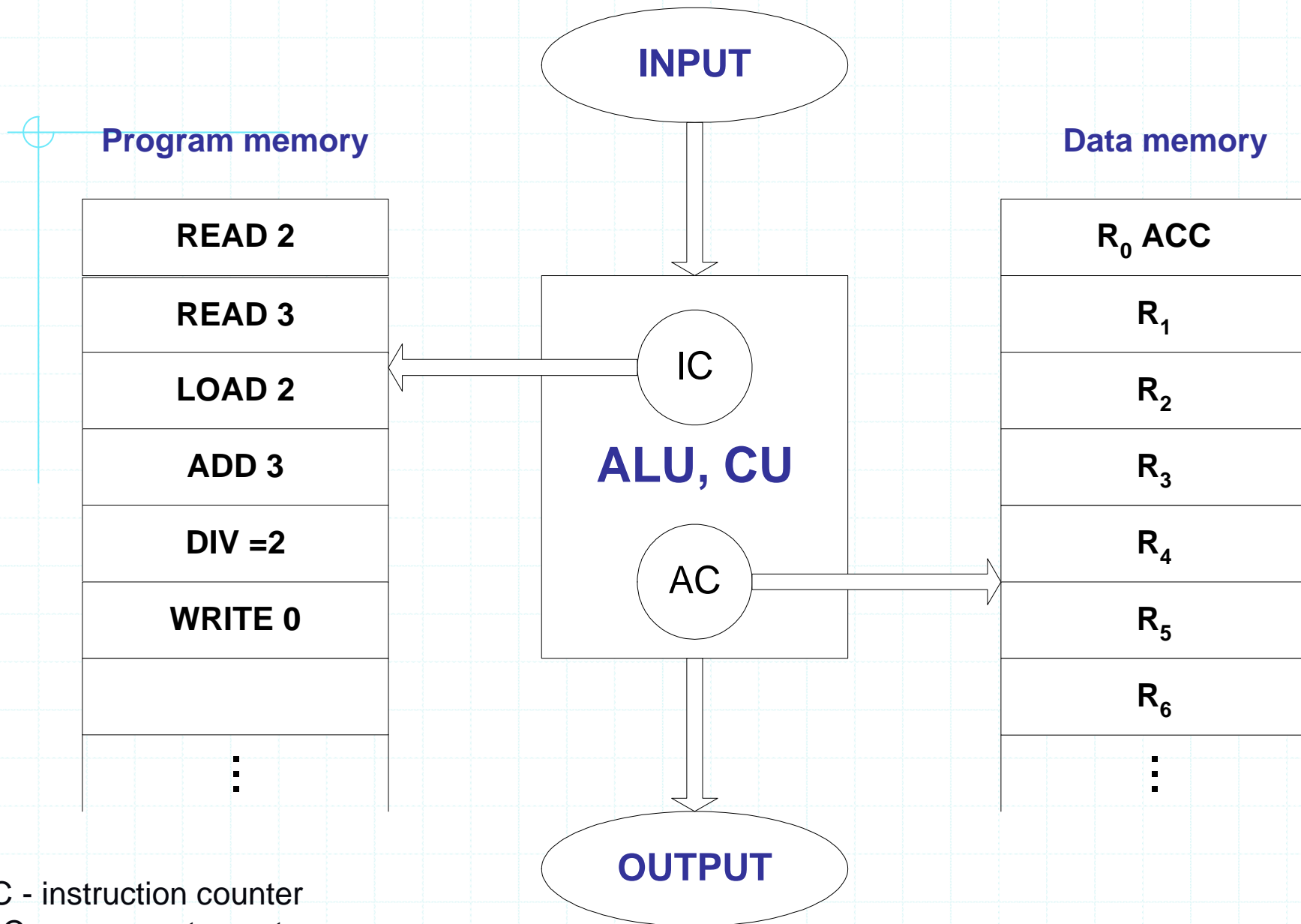
Teoretické základy informatiky

Random Access Machine

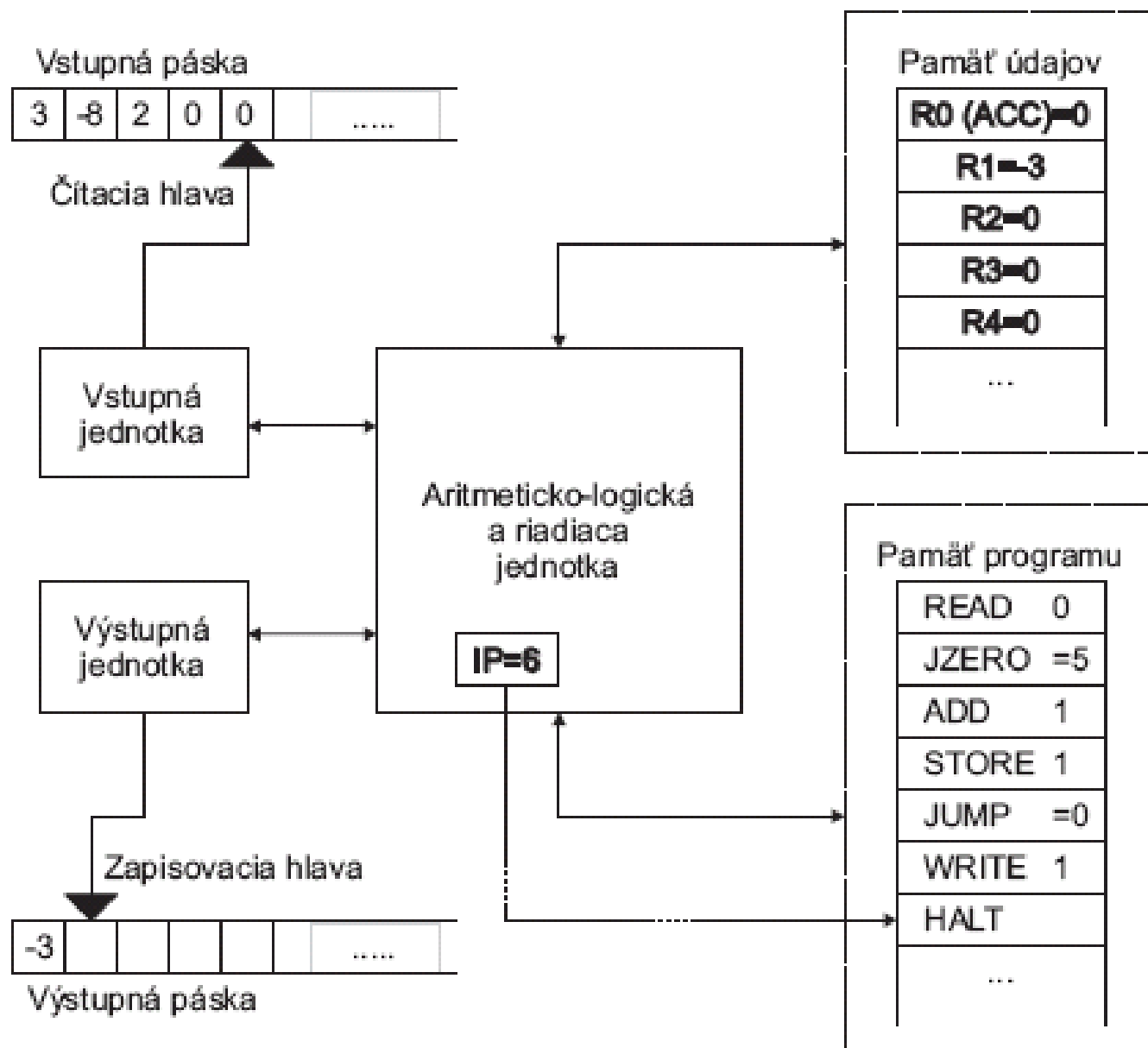
Random Access Machine

- výpočtový model RAM – stroj s ľubovoľným prístupom do pamäti
- model počítača, ktorý dokáže realizovať inštrukcie, podobný klasickým sekvenčným počítačom
- obsahuje:
 - pamäť dát, tvorenú registrami,
 - pamäť programu s inštrukciami,
 - aritmeticko-logickú jednotku,
 - rozhranie s okolím – vstupná a výstupná jednotka.
- vhodný výpočtový model na skúmanie algoritmov z hľadiska výpočtovej a priestorovej zložitosti

Model RAM - RANDOM ACCESS MACHINE



IC - instruction counter
AC – argument counter



Adresovanie RAM

- =i argument = konštanta i
- i obsah R_i
- *i nepriama adresácia, obsah registra, na ktorý ukazuje adresa v R_i

Adresovanie RAM, typy operandov

Operand	Príklad	Popis
=i	ADD =-5	Konštanta <i>i</i> . V príklade použitia sa k akumulátoru pričíta číslo 5.
i	ADD 4	Priame adresovanie. Operand sa vyhodnotí ako obsah registra <i>i</i> . V príklade použitia sa k akumulátoru pričíta obsah registra s indexom 4.
*i	ADD *7	Nepriame adresovanie. Obsah registra <i>i</i> určí register, ktorého obsah je výsledkom vyhodnotenia operandu. V príklade použitia sa k akumulátoru pričíta obsah registra s indexom daným obsahom registra číslo 7.

Inštrukcie v RAM

INŠTRUKCIA: MENO operand

Manipulácia s pamäťou:

READ op
WRITE op
LOAD op
STORE op

Výpočty:

ADD op
SUB op
MULT op
DIV op

Riadiace príkazy:

JUMP label
JZERO label
JGZERO label
HALT
ACCEPT
REJECT

Inštrukčná sada RAM

Inštrukcia		
Kód	Operand	Popis
Inštrukcie pre prácu s pamäťou		
LOAD	operand	operand sa načíta do akumulátora
STORE	operand	obsah akumulátora sa zapíše do pamäte podľa operandu
Aritmetické inštrukcie		
ADD	operand	operand sa pričíta do akumulátora (an. addition)
SUB	operand	operand sa odčíta od akumulátora (an. subtraction)
MUL	operand	akumulátora sa prenásobí operandom (an. multiplication)
DIV	operand	akumulátor sa predelý operandom (an. division)
Vstupno výstupné inštrukcie		
READ	operand	údaj zo vstupnej pásky sa zapíše do pamäte podľa operandu
WRITE	operand	operand sa zapíše na výstupnú pásku
Riadiace inštrukcie		
JUMP	návestie	skok na návestie
JZERO	návestie	skok na návestie, ak akumulátor nulový (an. jump if zero)
JGTZ	návestie	skok, ak akum. väčší ako nula (an. jump if grater than zero)
HALT		zastavenie výpočtu

Fungovanie zariadenia RAM

Dátová pamäť – nekonečne veľa registrov $R_1 \dots R_N$, dá sa uvažovať, že využívam konečnú množinu registrov, reálny počítač – uspokojivo veľká pamäť pre naše potreby

Akumulátor – špeciálny register R_0 , spolupracuje ALU, uchováva si výsledok

Pamäť programu, Realizácia inštrukcií – inštrukcie sa realizujú jedna za druhou

IC – instruction counter, instruction pointer, ukazuje na inštrukciu, ktorá sa aktuálne vykonáva

AC – argument counter, ukazuje na argument, ktorý sa spracováva

Sémantika (1)

READ op

číslo zo vstupu prenesie do op ($\text{input} \rightarrow R_{\text{op}}$)

WRITE op

to čo je určené v op na výstup ($R_{\text{op}} \rightarrow \text{output}$)

VSTUP/VÝSTUP

nekonečná páska len s číslami, číta čísla vcelku

Uchovávanie dát v registroch

LOAD op

z op zapíše do Acc ($\text{op} \rightarrow R_0$)

STORE op

z Acc zapíše do op ($R_0 \rightarrow \text{op}$)

Aritmetické operácie

ADD

$(R_0 + \text{op} \rightarrow R_0)$

SUB

$(R_0 - \text{op} \rightarrow R_0)$

R_0 je prvý

MULT

$(R_0 * \text{op} \rightarrow R_0)$

DIV

$(R_0 / \text{op} \rightarrow R_0)$

celočíselne, R_0 je prvý

Sémantika (2)

Skoky

JUMP label

nepodmienený skok na label GOTO label

JZERO label

podmienený if $R_0=0$ then GOTO label

JGZERO label

podmienený if $R_0>0$ then GOTO label

Zastavenie

HALT

zastavenie programu, bez návestia, ďalej sa nepokračuje

Modifikácie RAM

- zmena inštrukčnej sady

RAM+ stroj bez MUL a DIV

- stroj na rozpoznávanie jazykov

pridaním operácií ACCEPT a REJECT

- umožnením vlastnej modifikácie

RASP, môže modifikovať sám seba zmenou inštrukcií v pamäti, Random Access Stored program

PR.:Aritmetický priemer 2 hodnôt

READ 2

načítam prvý argument do R_2

READ 3

načítam prvý argument do R_3

LOAD 2

prvý argument umiestním do R_0

ADD 3

pripočítam obsah R_3

DIV =2

vydelím dvoma

WRITE 0

v ACC je výsledok, vypíšem ho na výstup

vstup R_2, R_3

výstup R_0

Príklad

Pomocou stroja RAM realizujte funkciu $f(n) = n!$, kde $n \in \mathbb{N}^+$

```
read(n), fak = 1  
for (i=1, i <= n, i++)  
{  
    fak = fak * i  
}  
write (fak)
```

Príklad

Pomocou stroja RAM realizujte funkciu $f(n) = n!$, kde $n \in \mathbb{N}^+$

```
read(n), fak = 1
for (i=n, i <= n, i--)
{
    fak = fak * i
}
write (fak)
```

Zac:

```
READ 1
LOAD 1
STORE 2
LOAD 1
SUB =1
STORE 1
JZERO End
MULT 2
STORE 2
JUMP Zac
WRITE 2
HALT
```

End:

Príklad

Pomocou stroja RAM realizujte funkciu $f(n) = 2^n$, kde $n \in \mathbb{N}^+$

```
read(n), R2 = 1
for (i=1, i <= n, i++)
{
  R2 = R2 * 2
}
write (R2)
```


Príklad

Pomocou stroja RAM realizujte funkciu $f(n) = 2^n$, kde $n \in \mathbb{N}^+$

```
read(n), R2 = 1
for (i=1, i <= n, i++)
{
  R2 = R2 * 2
}
write (R2)
```

Zac:

```
READ 1
LOAD =1
STORE 2
```

```
LOAD 1
JZERO End
LOAD 2
MULT =2
STORE 2
LOAD 1
SUB =1
STORE 1
JUMP Zac
```

End:

```
WRITE 2
HALT
```

Príklad

Pomocou stroja RAM realizujte funkciu $f(n) = 2^{2^n}$, kde $n \in \mathbb{N}^+$

```
read(n), R2 = 2
for (i=1, i <= n, i++)
{
    R2 = R2 * R2
}
write(R2)
```

Príklad

Pomocou stroja RAM realizujte funkciu $f(n) = 2^{2^n}$, kde $n \in \mathbb{N}^+$

```
read(n), R2 = 2
for (i=1, i <= n, i++)
{
    R2 = R2 * R2
}
write(R2)
```

Zac:

```
READ 1
LOAD =2
STORE 2
LOAD 1
JGZERO For
WRITE 2
HALT
```

For:

```
LOAD 2
MULT 2
STORE 2
LOAD 1
SUB =1
STORE 1
JUMP Zac
```

Zložitosť v modeli RAM

Výpočtová zložitosť v modeli RAM:

- časová
- priestorová, pamäťová

Zložitosťné miery na RAM:

- jednotková zložitosťná miera
- logaritmická zložitosťná miera

RAM - Cena inštrukcií

- Jednotková cena
cena každej inštrukcie je rovná 1
- Logaritmická cena
Nech $l(i)$ je logaritmická funkcia čísla daná takto:
$$l(i) = \lfloor \log_2 i \rfloor + 1; i \neq 0$$
$$l(i) = 1; i = 0$$

Časová náročnosť práce s jednotlivými operadami

Operand op	Cena $t(op)$	Popis
$=i$	$l(i)$	Cena daná náročnosťou manipulácie s číslom i .
i	$l(i) + l(c(i))$	Cena daná náročnosťou manipulácie s indexom registra i (čím vyšší index, tým nákladnejší prístup) a obsahom registra i .
$*i$	$l(i) + l(c(i)) + l(c(c(i)))$	Je nevyhnutné „zaplatiť“ za prístup do registra i , potom za prístup do registra $c(i)$ (register je určený nepriamou adresáciou pomocou registra i) a nakoniec za manipuláciu s číslom $c(c(i))$ v registri $c(i)$.

Časová náročnosť jednotlivých inštrukcií

Inštrukcia		
Kód	Operand	Cena
LOAD	op	$t(op)$
STORE	i	$l(c(0)) + l(i)$
STORE	*i	$l(c(0)) + l(i) + l(c(i))$
ADD	op	$l(c(0)) + t(op)$
SUB	op	$l(c(0)) + t(op)$
MUL	op	$l(c(0)) + t(op)$
DIV	op	$l(c(0)) + t(op)$
READ	i	$l(vstup) + l(i)$
READ	*i	$l(vstup) + l(i) + l(c(i))$
WRITE	op	$t(op)$
JUMP	návestie	1
JZERO	návestie	$l(c(0))$
JGTZ	návestie	$l(c(0))$
HALT		1

Pamäťová zložitosť S programu P na počítači RAM sa rovná súčet maximálnych cien všetkých vykonaných inštrukcií cez všetky použité registre

$$S = \sum_{R_j \in \mathbb{R}} \max_{ins \in V(P)} Cena(ins R_j)$$

Časová zložitosť T programu P na počítači RAM sa rovná súčtu cien všetkých vykonaných inštrukcií počas výpočtu programu P

$$T = \sum_{ins \in V(P)} Cena(ins)$$

Sumarizácia zložitostných mier na stroji RAM

Typ zložitosti	Zložitostná miera	
	Jednotková	Logaritmická
Časová	Jednotková časová	Logaritmická časová
Priestorová	Jednotková priestorová	Logaritmická priestorová

Príklad

$$f(x) = 2^N$$

```
read(N),  
R2 = 1  
for (i=N, i >= 0, i- -)  
{  
  R2 = R2 * 2  
}  
write (R2)
```

```
R1 ← N  
R2 ← 2N
```

```
0      READ   1  
1      LOAD   =1  
2      STORE  2  
3      LOAD   1  
4      JZERO  =12  
5      LOAD   2  
6      MUL    =2  
7      STORE  2  
8      LOAD   1  
9      SUB    =1  
10     STORE  1  
11     JUMP    =3  
12     WRITE  2  
13     HALT
```

```
0      READ   N  
1      LOAD   =1  
2      STORE  RES  
3      next:  LOAD  N  
4          JZERO  end  
5      LOAD   RES  
6      MUL    =2  
7      STORE  RES  
8      LOAD   N  
9      SUB    =1  
10     STORE  N  
11     JUMP    next  
12     end:   WRITE RES  
13     HALT
```

Inštrukcia			Počet vykonaní	Logaritmická cena
	READ	1	1	$l(n) + l(1)$
	LOAD	=1	1	$l(1)$
	STORE	2	1	$l(2) + l(1)$
next:	LOAD	1	$n+1$	$l(1) + l(n-i)$
	JZERO	end	$n+1$	$l(n-i)$
	LOAD	2	n	$l(2) + l(2^i)$
	MUL	=2	n	$l(2^i) + l(2)$
	STORE	2	n	$l(2^{i+1}) + l(2)$
	LOAD	1	n	$l(1) + l(n-i)$
	SUB	=1	n	$l(n-i) + l(1)$
	STORE	1	n	$l(n-i-1) + l(1)$
	JUMP	next	n	1
end:	WRITE	2	1	$l(2) + l(2^n)$
	HALT		1	1

← (1)

← (2)

← (3)

$$T_1(n) = 7 + 9n \text{ čiže } T_1(n) = O(n).$$

$$S_1(n) = 3, \text{ čiže } S_1(n) = O(1).$$

$$S_l(n) = 2l(2^n) + l(n) \approx n \log 2 = O(n).$$

$$(1) \quad T_l^A(n) = l(n) + 3l(1) + l(2) = O(\log(n))$$

$$(2) \quad T_l^B(n) \approx \sum_{i=0}^{n-1} \log(2^i) + \log(n) = O(n^2)$$

$$(3) \quad T_l^C(n) = 1 + l(2) + l(2^n) \approx \log 2^n = n \log 2 = O(n)$$

Pravidlá pri výpočte zložitosti

- Linearita sumácie

$$\sum_i (c_1 \cdot a_i + c_2 \cdot b_i) = c_1 \cdot \sum_i a_i + c_2 \cdot \sum_i b_i$$

- Súčet členov aritmetickej postupnosti

$$\sum_{i=1}^n a_i = \frac{n}{2}(a_1 + a_n); a_{i+1} = a_i + d$$

- Súčet členov geometrickej postupnosti

$$\sum_{i=1}^n a_i = a_1 \frac{q^n - 1}{q - 1}; a_{i+1} = a_i \cdot q$$

- Vlastnosti logaritmov

$$\log A + \log B = \log(A \cdot B)$$

$$\log A^B = B \cdot \log A$$

$$\log_a X = \frac{\log_c X}{\log_c a}$$

Výpočtová zložitosť príkladu $f(x)=2^N$ na modeli RAM

Inštrukcia			Počet vykonaní	Logaritmická cena
	READ	1	1	$l(n) + l(1)$
	LOAD	=1	1	$l(1)$
	STORE	2	1	$l(2) + l(1)$
next:	LOAD	1	$n+1$	$l(1) + l(n - i)$
	JZERO	end	$n+1$	$l(n - i)$
	LOAD	2	n	$l(2) + l(2^i)$
	MUL	=2	n	$l(2^i) + l(2)$
	STORE	2	n	$l(2^{i+1}) + l(2)$
	LOAD	1	n	$l(1) + l(n - i)$
	SUB	=1	n	$l(n - i) + l(1)$
	STORE	1	n	$l(n - i - 1) + l(1)$
	JUMP	next	n	1
end:	WRITE	2	1	$l(2) + l(2^n)$
	HALT		1	1

Typ zložitosti	Zložitosťná miera	
	Jednotková	Logaritmická
Časová	$T_1(n) = O(n)$	$T_l(n) = O(n^2)$
Priestorová	$S_1(n) = O(1)$	$S_l(n) = O(n)$

$$\underline{f(n) = 2^{2^n}}$$

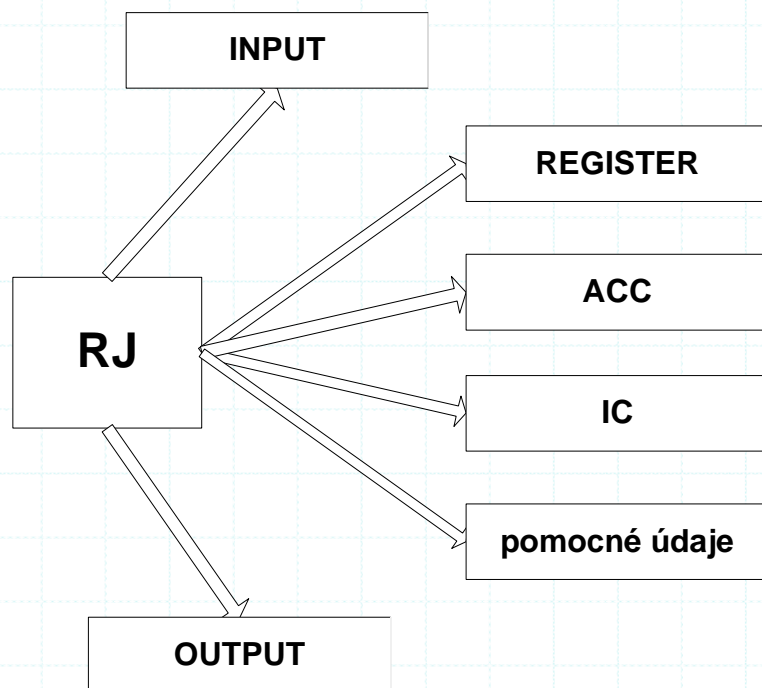
	READ	N	$(N \leftarrow n)$
	LOAD	=2	
while:	STORE	temp	$(temp \leftarrow 2^{2^{n-N}})$
	LOAD	N	
	JGZERO	body	$(while\ N > 0\ do)$
	WRITE	temp	
	HALT		
body:	SUB	=1	
	STORE	N	$(N \leftarrow N - 1)$
	LOAD	temp	
	MULT	0	$(R_0 \leftarrow temp^2)$
	JUMP	while	

Analýza složitosti programu

Jednotková cena	Logaritmická cena
Paměť	Paměť
$S(n) = O(1)$	$S(n) = O(2^n)$
Paměť	Paměť
$T(n) = O(2^{\log n})$	$T(n) = O(2^n)$

Ekvivalencia RAM \Leftrightarrow TS

Simulácia RAM na TS – 6 páskový TS, simuluje sa každá inštrukcia zvlášť podľa polohy IC



Ekvivalencia výpočtových modelov

Veta 6.4.1 (*O ekvivalenciách výpočtových modeloch*) Nasledujúce výpočtové modely sú ekvivalentné:

1. *Turingov stroj*
2. *Počítadlový stroj*
3. *Stroj RAM*

Abstraktné zložitosťné triedy

Uvažujme k -páskové (časovo / páskovo) ohraničené Turingove stroje so vstupným slovom w takým, že platí: $|w| = n$.

$DTIME(f(n))$ - trieda jazykov rozpoznávaných deterministickým Turingovým strojom s časovým ohraničením max. $f(|w|)$.

$NTIME(f(n))$ - trieda jazykov rozpoznávaných nedeterministickým Turingovým strojom s časovým ohraničením max. $f(|w|)$.

$DSPACE(g(n))$ - trieda jazykov rozpoznávaných deterministickým Turingovým strojom s časovým ohraničením max. $g(|w|)$.

$NSPACE(g(n))$ - trieda jazykov rozpoznávaných nedeterministickým Turingovým strojom s časovým ohraničením max. $g(|w|)$.

$$P = \bigcup_{k=0}^{\infty} DTIME(n^k)$$

$$NP = \bigcup_{k=0}^{\infty} NTIME(n^k)$$

Ďakujem za pozornosť.