

Matematika v PLC

Michal Kvasnica

Riešenie kvadratickej rovnice

- V Matlabe:
 - $a = -1; b = -1; c = 6;$
 - $D = b*b - 4*a*c = 25$
 - $x1 = (-b + \text{sqrt}(D))/(2*a) = -3$
 - $x2 = (-b - \text{sqrt}(D))/(2*a) = 2$

Riešenie kvadratickej rovnice

- Majme rovnicu $ax^2 + bx + c = 0$
- Chceme nájsť jej dve riešenia x_1 a x_2

Riešenie kvadratickej rovnice

- V Matlabe:
 - $a = -1; b = -1; c = 6;$
 - $D = b*b - 4*a*c = 25$
 - $x1 = (-b + \text{sqrt}(D))/(2*a) = -3$
 - $x2 = (-b - \text{sqrt}(D))/(2*a) = 2$
- Čo Matlab urobí za nás?

Riešenie kvadratickej rovnice

- V Matlabe:
 - $a = -1$; $b = -1$; $c = 6$;
 - $D = b*b - 4*a*c$
- Tu Matlab spraví dve veci:
 - vytvorí v pamäti miesto pre premennú "a"
 - do vytvorenej pamäťovej bunky uloží číslo -1

Riešenie kvadratickej rovnice

- V Matlabe:
 - $a = -1$; $b = -1$; $c = 6$;
 - $D = b*b - 4*a*c$

Riešenie kvadratickej rovnice

- V Matlabe:
 - $a = -1$; $b = -1$; $c = 6$;
 - $D = b*b - 4*a*c$
- Tu Matlab spraví päť vecí:
 - vytvorí v pamäti miesto pre premennú "D"

Riešenie kvadratickej rovnice

- V Matlabe:
 - $a = -1$; $b = -1$; $c = 6$;
 - $D = b*b - 4*a*c$
- Tu Matlab spraví päť vecí:
 - vytvorí v pamäti miesto pre premennú "D"
 - vypočíta "b*b" a výsledok si odloží niekam do pamäte (napr. do internej premennej P1)

Riešenie kvadratickej rovnice

- V Matlabe:
 - $a = -1; b = -1; c = 6;$
 - $D = b*b - 4*a*c$
- Tu Matlab spraví päť vecí:
 - vytvorí v pamäti miesto pre premennú "D"
 - vypočíta "b*b" a výsledok si odloží niekam do pamäte (napr. do internej premennej P1)
 - vypočíta "4*a" a výsledok zasa uloží niekam do pamäte (povedzme do internej premennej P2)

Riešenie kvadratickej rovnice

- V Matlabe:
 - $a = -1; b = -1; c = 6;$
 - $D = b*b - 4*a*c$
- Tu Matlab spraví päť vecí:
 - vytvorí v pamäti miesto pre premennú "D"
 - vypočíta "b*b" a výsledok si odloží niekam do pamäte (napr. do internej premennej P1)
 - vypočíta "4*a" a výsledok zasa uloží niekam do pamäte (povedzme do internej premennej P2)
 - vypočíta "P2*c" a zasa si výsledok niekam uloží (P3)
 - vypočíta P1 - P3 a výsledok uloží do premennej D

Riešenie kvadratickej rovnice

- V Matlabe:
 - $a = -1; b = -1; c = 6;$
 - $D = b*b - 4*a*c$
- Tu Matlab spraví päť vecí:
 - vytvorí v pamäti miesto pre premennú "D"
 - vypočíta "b*b" a výsledok si odloží niekam do pamäte (napr. do internej premennej P1)
 - vypočíta "4*a" a výsledok zasa uloží niekam do pamäte (povedzme do internej premennej P2)
 - vypočíta "P2*c" a zasa si výsledok niekam uloží (P3)

Zhrnutie

- Akákoľvek zložitejšia matematická operácia sa v skutočnosti interne vyhodnocuje po jednotlivých základných krokoch (sčítovanie, odčítovanie, násobenie, odmocňovanie, atď)
- Každý čiastkový výsledok sa musí uložiť do pamäte
- V pamäti sa musí vytvoriť miesto pre jednotlivé premenné (aj tie, do ktorých sa vkladajú čiastkové výpočty!)

Vytvorenie pamäťových miest

- Čísla s desatinnou čiarkou sú uložené vo formáte **double word**, čo je 32 bitov, čiže 4 bajty

VD100	31	VB100	24	23	VB101	16	15	VB102	8	7	VB103	0
-------	----	-------	----	----	-------	----	----	-------	---	---	-------	---

- Formát adresy pamäťového miesta: VDX
 - V = interná premenná
 - D = double word
 - X = číslo začiatočného bajtu (napr. 100)

Vytvorenie pamäťových miest

- Vytvorme pamäťové miesta pre double word premenné "a", "b", "c", "P1" (výsledok $b*b$), "P2" ($4*a$), "P3" ($P2*c$), a "D" ($P1 - P3$)
- a: VD0 (obsadzuje bajty č. 0, 1, 2 a 3)
- b: VD???

Vytvorenie pamäťových miest

- Vytvorme pamäťové miesta pre double word premenné "a", "b", "c", "P1" (výsledok $b*b$), "P2" ($4*a$), "P3" ($P2*c$), a "D" ($P1 - P3$)

Vytvorenie pamäťových miest

- Vytvorme pamäťové miesta pre double word premenné "a", "b", "c", "P1" (výsledok $b*b$), "P2" ($4*a$), "P3" ($P2*c$), a "D" ($P1 - P3$)
- a: VD0 (obsadzuje bajty č. 0, 1, 2 a 3)
- b: VD4 (prvé voľné miesto po VD0)

Vytvorenie pamäťových miest

- Vytvorme pamäťové miesta pre double word premenné "a", "b", "c", "P1" (výsledok $b*b$), "P2" ($4*a$), "P3" ($P2*c$), a "D" ($P1 - P3$)
- a: VD0 (obsadzuje bajty č. 0, 1, 2 a 3)
- b: VD4 (prvé voľné miesto po VD0)
- c: VD

Vytvorenie pamäťových miest

- Vytvorme pamäťové miesta pre double word premenné "a", "b", "c", "P1" (výsledok $b*b$), "P2" ($4*a$), "P3" ($P2*c$), a "D" ($P1 - P3$)
- a: VD0 (obsadzuje bajty č. 0, 1, 2 a 3)
- b: VD4 (prvé voľné miesto po VD0)
- c: VD8
- P1: VD12
- P2: VD16
- P3: VD20
- D: VD24

Vytvorenie pamäťových miest

- Vytvorme pamäťové miesta pre double word premenné "a", "b", "c", "P1" (výsledok $b*b$), "P2" ($4*a$), "P3" ($P2*c$), a "D" ($P1 - P3$)
- a: VD0 (obsadzuje bajty č. 0, 1, 2 a 3)
- b: VD4 (prvé voľné miesto po VD0)
- c: VD8

Postup pri PLC

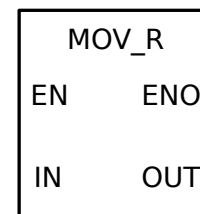
1. Vytvorenie pamäťových miest
2. Naplnenie pamäťových miest číslami
3. Vytvorenie rebríkového diagramu z matematických blokov

Naplnenie pamäte číslami

- Máme premenné a, b, c, P1, P2, P3, D
- Chceme priradiť a = -1, b = -1, c = 6

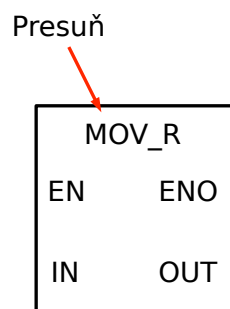
Naplnenie pamäte číslami

- Máme premenné a, b, c, P1, P2, P3, D
- Chceme priradiť a = -1, b = -1, c = 6



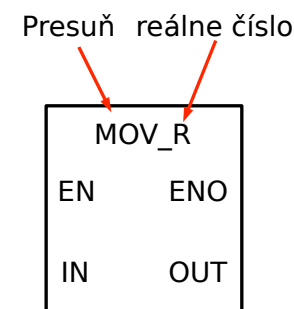
Naplnenie pamäte číslami

- Máme premenné a, b, c, P1, P2, P3, D
- Chceme priradiť a = -1, b = -1, c = 6



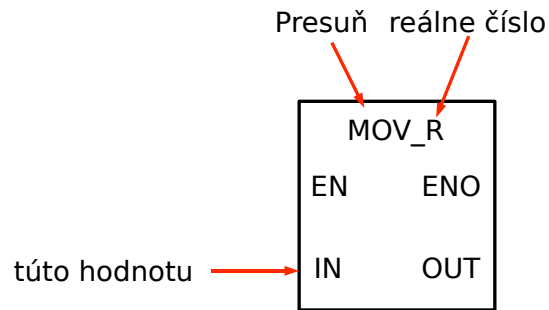
Naplnenie pamäte číslami

- Máme premenné a, b, c, P1, P2, P3, D
- Chceme priradiť a = -1, b = -1, c = 6



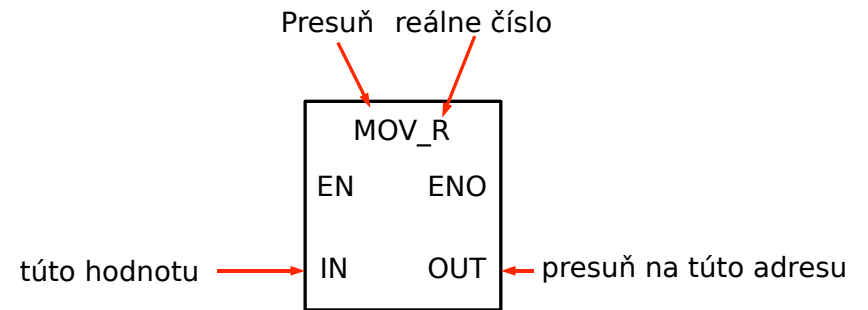
Napĺňanie pamäte islami

- Mame premenn a, b, c, P1, P2, P3, D
- Chceme priradiť a = -1, b = -1, c = 6



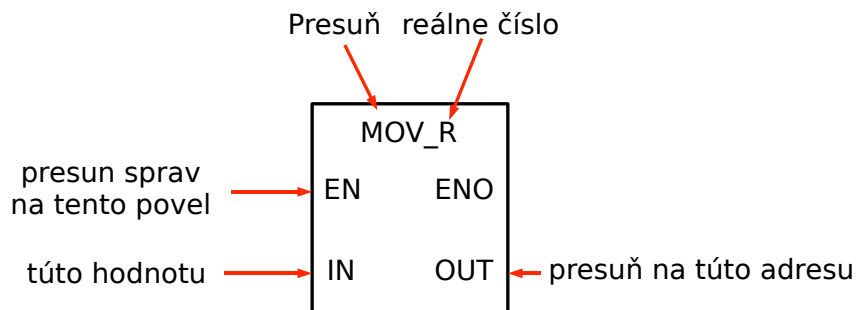
Napĺňanie pamäte islami

- Mame premenn a, b, c, P1, P2, P3, D
- Chceme priradiť a = -1, b = -1, c = 6



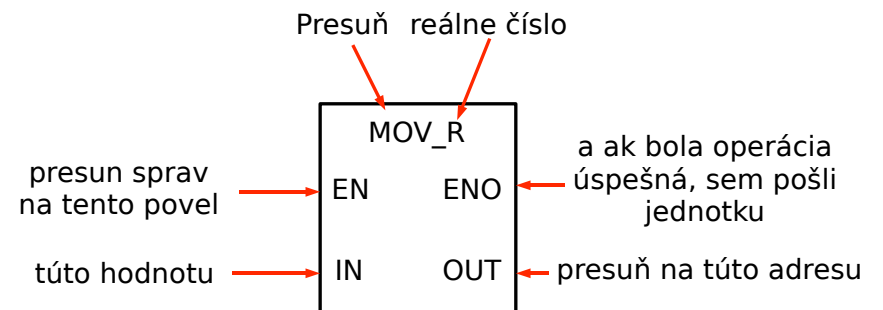
Napĺňanie pamäte islami

- Mame premenn a, b, c, P1, P2, P3, D
- Chceme priradiť a = -1, b = -1, c = 6



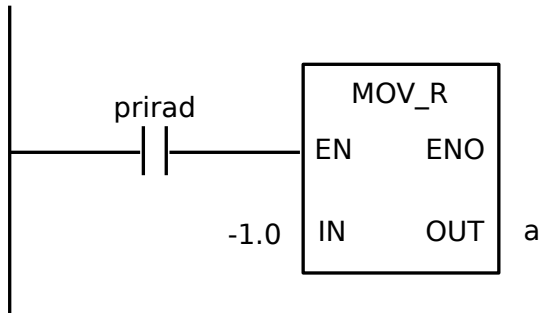
Napĺňanie pamäte islami

- Mame premenn a, b, c, P1, P2, P3, D
- Chceme priradiť a = -1, b = -1, c = 6



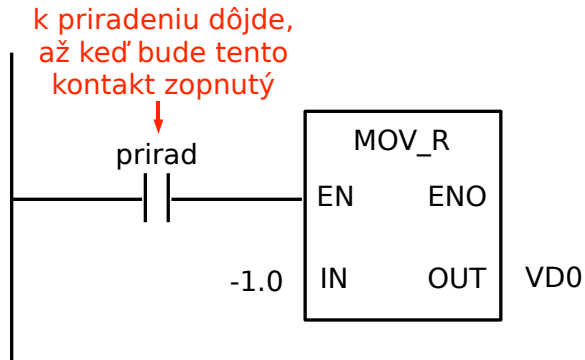
Príklad

- Máme premenné a, b, c, P1, P2, P3, D
- Chceme priradiť $a = -1$, $b = -1$, $c = 6$



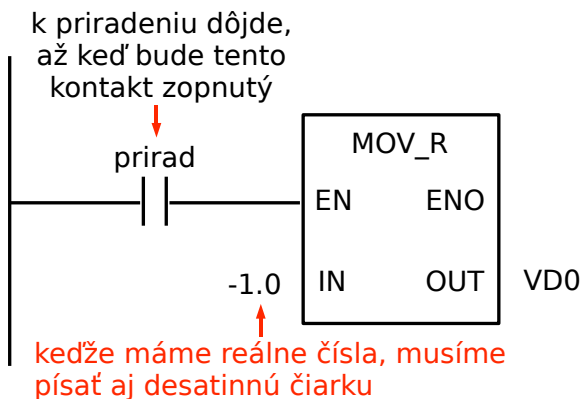
Príklad

- Máme premenné a, b, c, P1, P2, P3, D
- Chceme priradiť $a = -1$, $b = -1$, $c = 6$



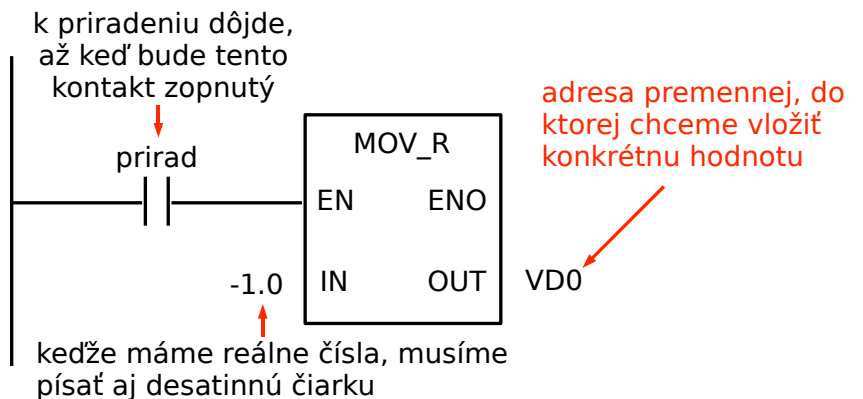
Príklad

- Máme premenné a, b, c, P1, P2, P3, D
- Chceme priradiť $a = -1$, $b = -1$, $c = 6$



Príklad

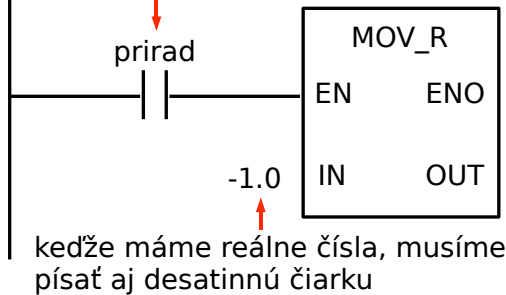
- Máme premenné a, b, c, P1, P2, P3, D
- Chceme priradiť $a = -1$, $b = -1$, $c = 6$



Príklad

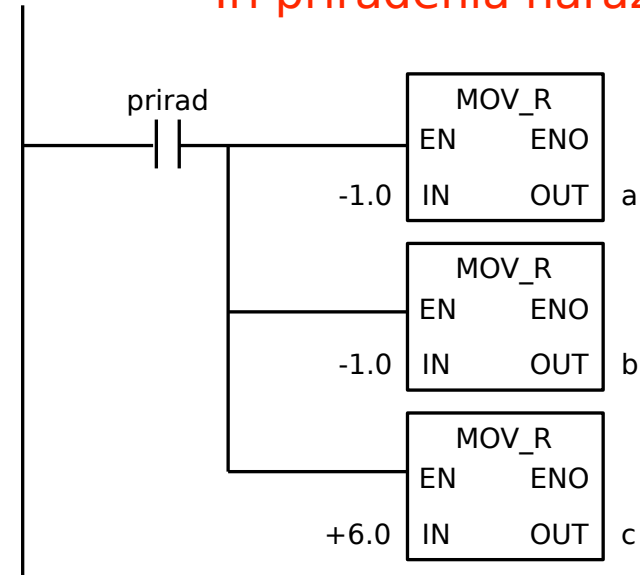
- Máme premenné a, b, c, P1, P2, P3, D
- Chceme priradiť $a = -1$, $b = -1$, $c = 6$

k priradeniu dôjde,
až keď bude tento
kontakt zopnutý



môžeme použiť aj
symbolické meno
premennej, ak sme ho
predtým priradili k
adrese VD0

Tri priradenia naraz



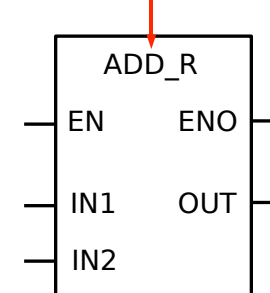
Postup pri PLC

1. Vytvorenie pamäťových miest
2. Naplnenie pamäťových miest číslami
3. Vytvorenie rebríkového diagramu z matematických blokov

Matematické bloky v PLC

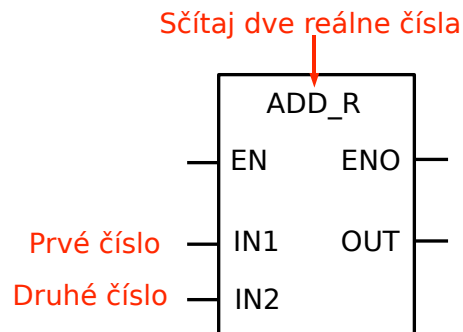
- Základné matematické operácie:
 - sčítanie (ADD_R), odčítanie (SUB_R), násobenie (MUL_R), delenie (DIV_R)
- Formát takýchto blokov:

Sčítaj dve reálne čísla



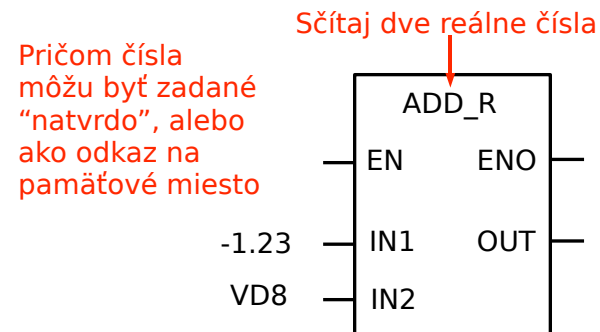
Matematické bloky v PLC

- Základné matematické operácie:
 - sčítanie (ADD_R), odčítanie (SUB_R), násobenie (MUL_R), delenie (DIV_R)
- Formát takýchto blokov:



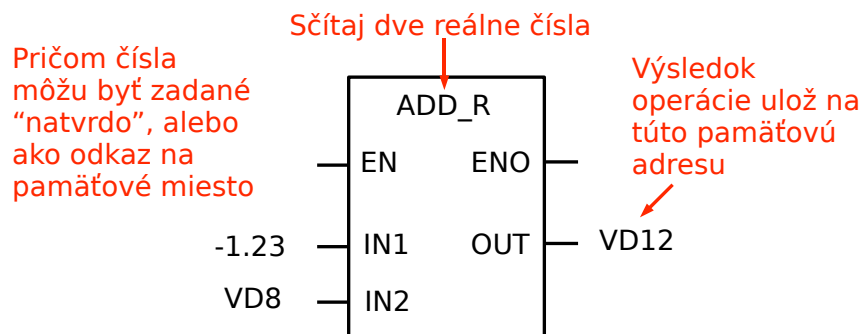
Matematické bloky v PLC

- Základné matematické operácie:
 - sčítanie (ADD_R), odčítanie (SUB_R), násobenie (MUL_R), delenie (DIV_R)
- Formát takýchto blokov:



Matematické bloky v PLC

- Základné matematické operácie:
 - sčítanie (ADD_R), odčítanie (SUB_R), násobenie (MUL_R), delenie (DIV_R)
- Formát takýchto blokov:

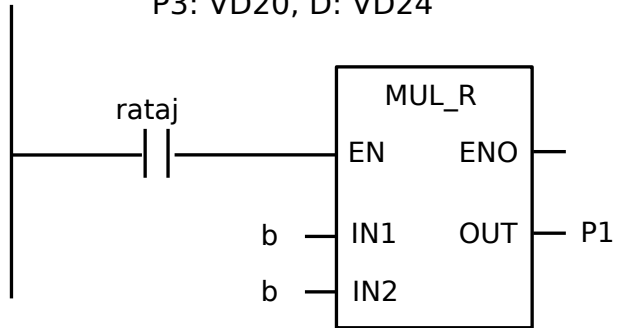


Výpočet diskriminantu

- $D = b*b - 4*a*c$
- $P1 = b*b$, $P2 = 4*a$, $P3 = P2*c$, $D = P1 - P3$
- Rozdelenie adries:
 - a: VD0, b: VD4, c: VD8, P1: VD12, P2: VD16, P3: VD20, D: VD24

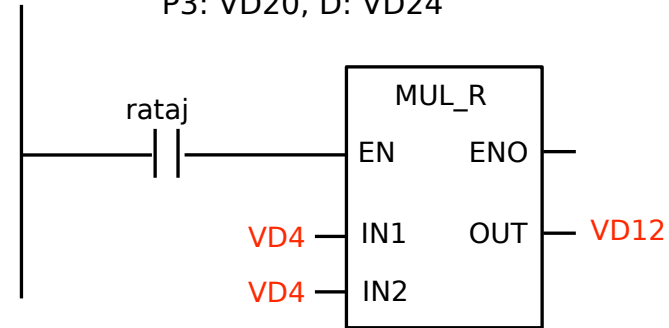
Výpočet diskriminantu

- $D = b*b - 4*a*c$
- $P1 = b*b$, $P2 = 4*a$, $P3 = P2*C$, $D = P1 - P3$
- Rozdelenie adries:
 - a: VD0, b: VD4, c: VD8, P1: VD12, P2: VD16, P3: VD20, D: VD24



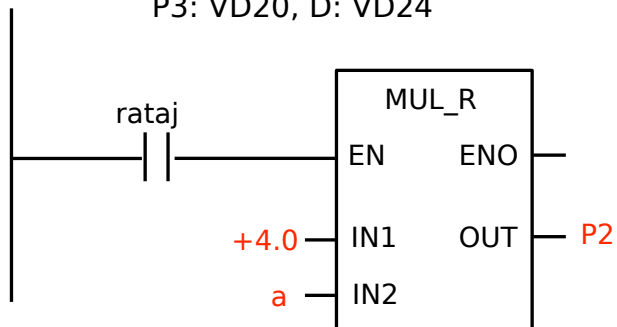
Alebo pomocou adries...

- $D = b*b - 4*a*c$
- $P1 = b*b$, $P2 = 4*a$, $P3 = P2*C$, $D = P1 - P3$
- Rozdelenie adries:
 - a: VD0, b: VD4, c: VD8, P1: VD12, P2: VD16, P3: VD20, D: VD24



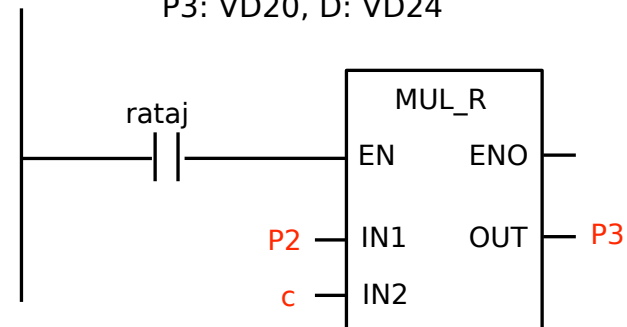
Pod to umiestnime ďalší blok

- $D = b*b - 4*a*c$
- $P1 = b*b$, $P2 = 4*a$, $P3 = P2*C$, $D = P1 - P3$
- Rozdelenie adries:
 - a: VD0, b: VD4, c: VD8, P1: VD12, P2: VD16, P3: VD20, D: VD24



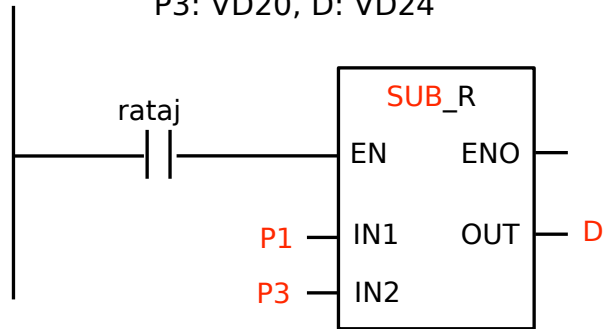
A pod to ďalší...

- $D = b*b - 4*a*c$
- $P1 = b*b$, $P2 = 4*a$, $P3 = P2*C$, $D = P1 - P3$
- Rozdelenie adries:
 - a: VD0, b: VD4, c: VD8, P1: VD12, P2: VD16, P3: VD20, D: VD24



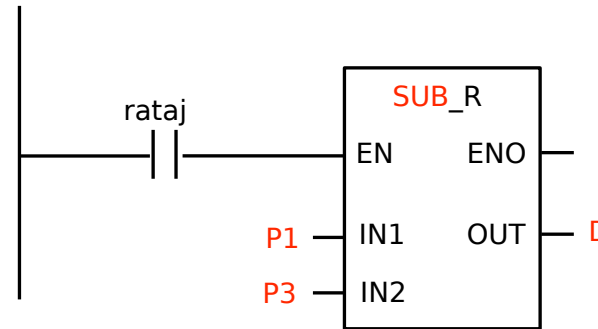
A pod to posledný...

- $D = b*b - 4*a*c$
- $P1 = b*b$, $P2 = 4*a$, $P3 = P2*c$, $D = P1 - P3$
- Rozdelenie adries:
 - a: VD0, b: VD4, c: VD8, P1: VD12, P2: VD16, P3: VD20, D: VD24



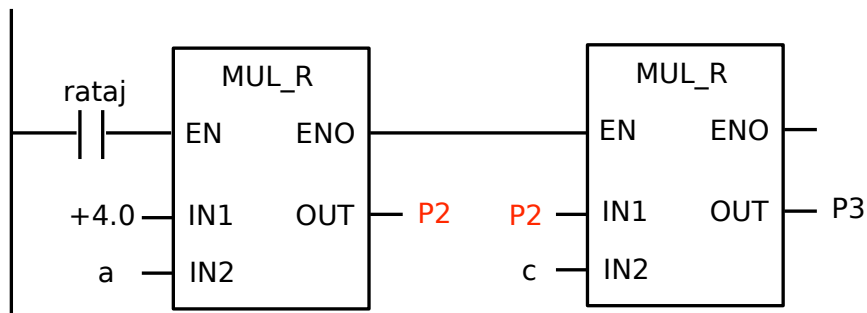
A pod to posledný...

- Samozrejme že záleží na poradí vstupných argumentov!
- SUB_R blok vždy vykoná operáciu IN1-IN2



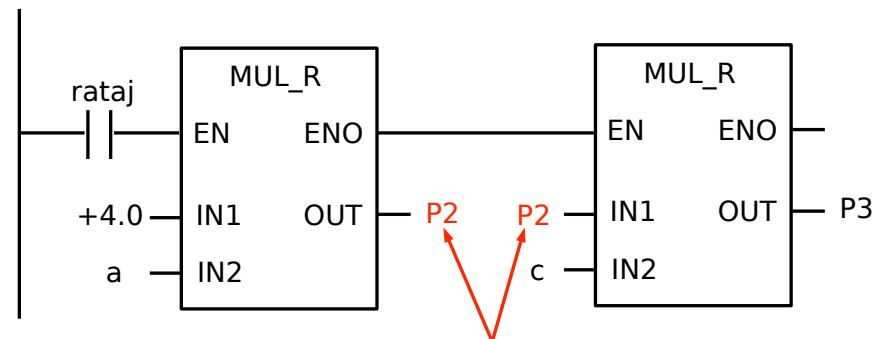
Alternatívne zapojenie

- Bloky môžeme zapojiť aj za sebou, využívame pri tom fakt, že ENO=1 ak operácia prebehla bez chýb



Alternatívne zapojenie

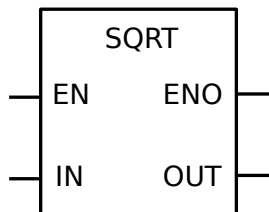
- Bloky môžeme zapojiť aj za sebou, využívame pri tom fakt, že ENO=1 ak operácia prebehla bez chýb



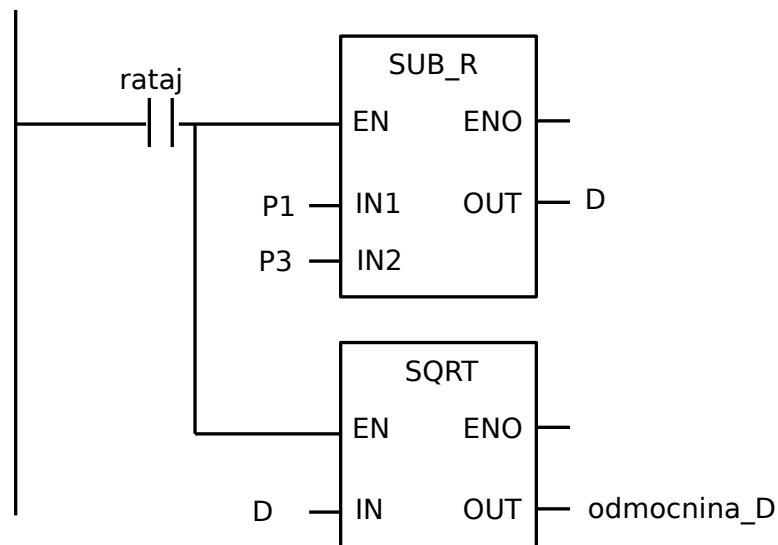
Hodnoty nie je možné prepájať čiarou!

Matematické bloky v PLC

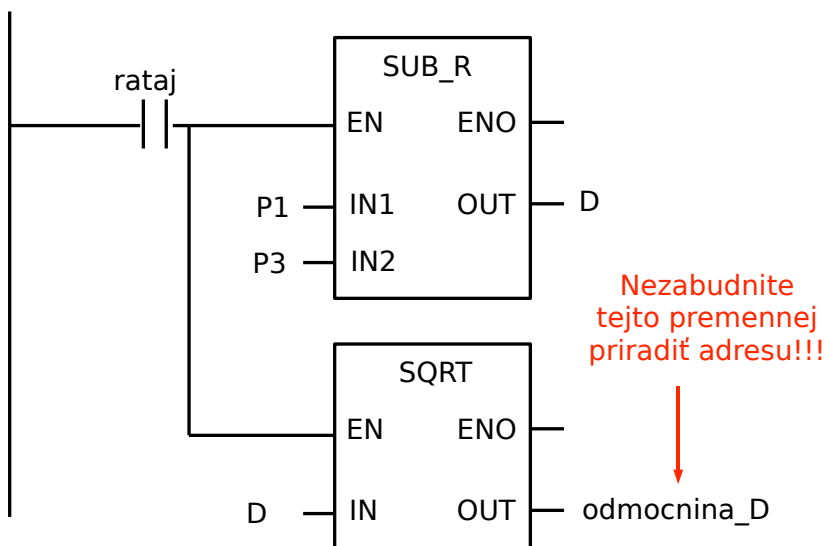
- Ostatné matematické operácie:
 - odmocňovanie (SQRT), trigonometrické funkcie (SIN, COS, TAN), prirodzený exponent (EX), prirodzený logaritmu (LN)
- Majú iba 2 vstupy:



Odmocnina diskriminantu



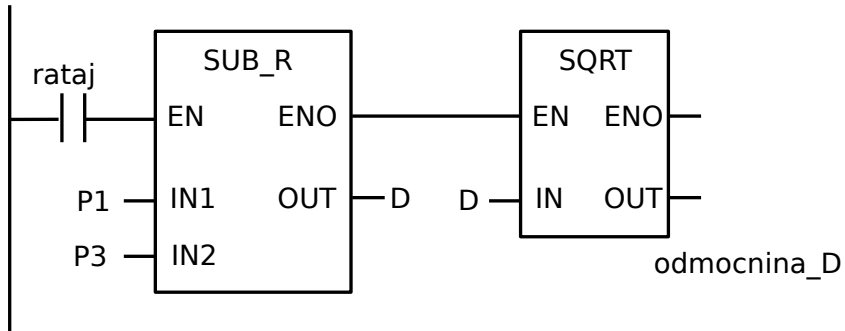
Odmocnina diskriminantu



Vytvorenie pamäťových miest

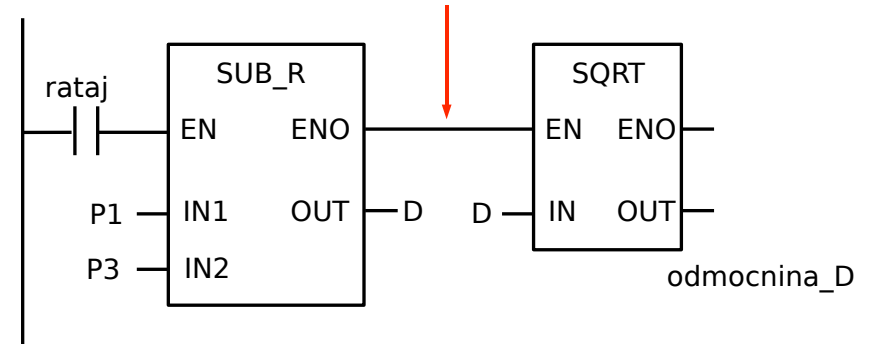
- a: VD0
- b: VD4
- c: VD8
- P1: VD12
- P2: VD16
- P3: VD20
- D: VD24
- odmocnina_D: VD28

Alternatívne zapojenie



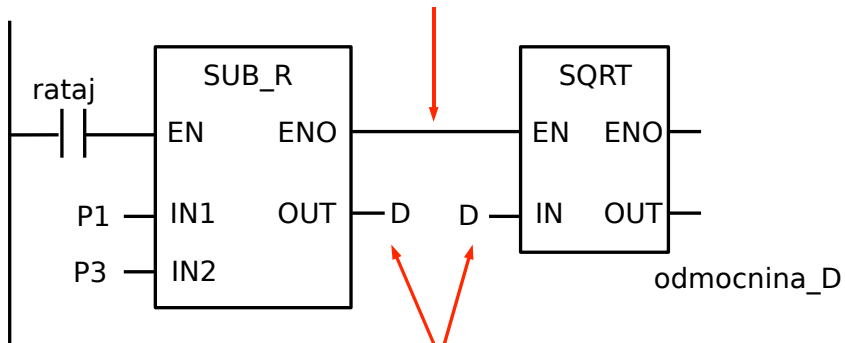
Alternatívne zapojenie

Po odčítaní sa automaticky spustí odmocňovanie



Alternatívne zapojenie

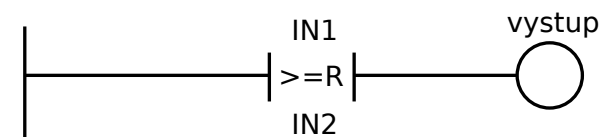
Po odčítaní sa automaticky spustí odmocňovanie



Hodnoty nie je možné prepájať čiarou!
Matematické bloky spracúvajú buď iba "natvrdo"
vložené čísla, alebo pamäťové adresy

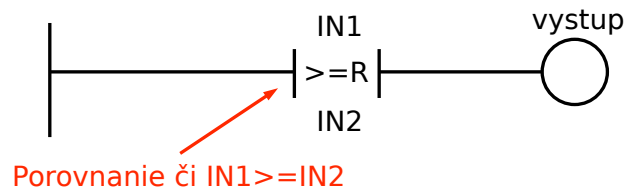
Porovnávacie operácie

- Čo sa stane ak číslo pod odmocninou bude záporné? Nič dobré...
- Ako overiť, že číslo pod odmocninou nie je záporné?



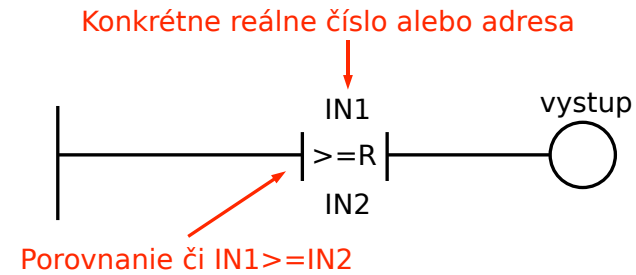
Porovnávacie operácie

- Čo sa stane ak číslo pod odmocninou bude záporné? Nič dobré...
- Ako overiť, že číslo pod odmocninou nie je záporné?



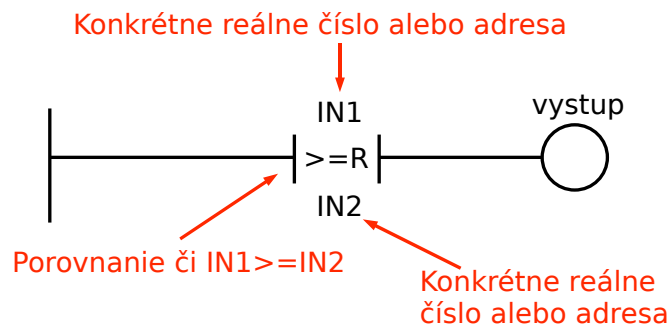
Porovnávacie operácie

- Čo sa stane ak číslo pod odmocninou bude záporné? Nič dobré...
- Ako overiť, že číslo pod odmocninou nie je záporné?



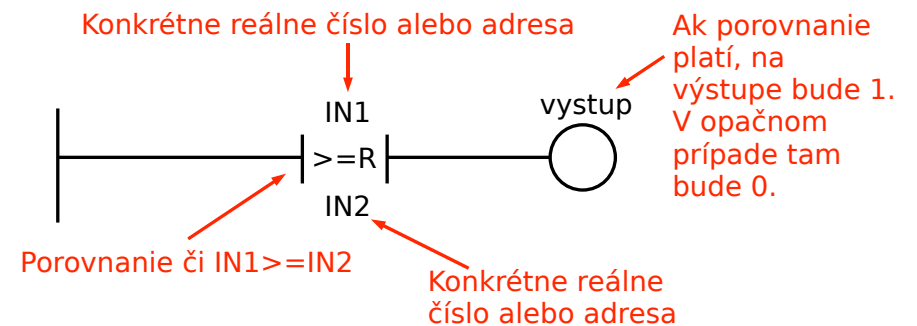
Porovnávacie operácie

- Čo sa stane ak číslo pod odmocninou bude záporné? Nič dobré...
- Ako overiť, že číslo pod odmocninou nie je záporné?

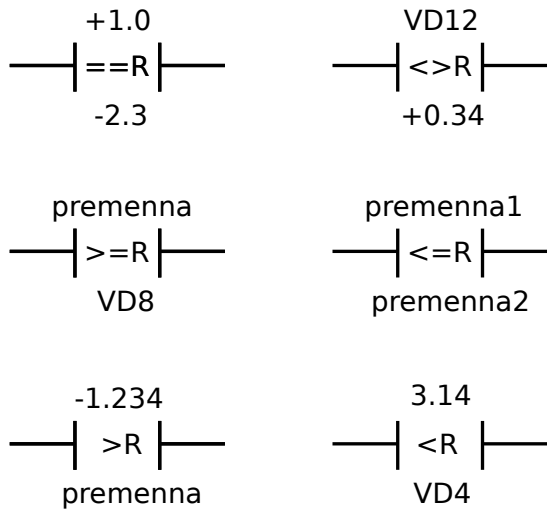


Porovnávacie operácie

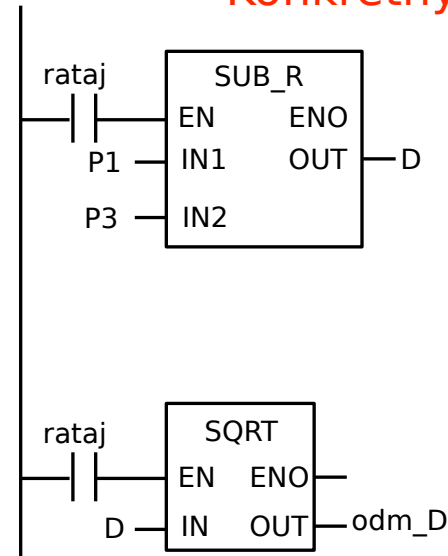
- Čo sa stane ak číslo pod odmocninou bude záporné? Nič dobré...
- Ako overiť, že číslo pod odmocninou nie je záporné?



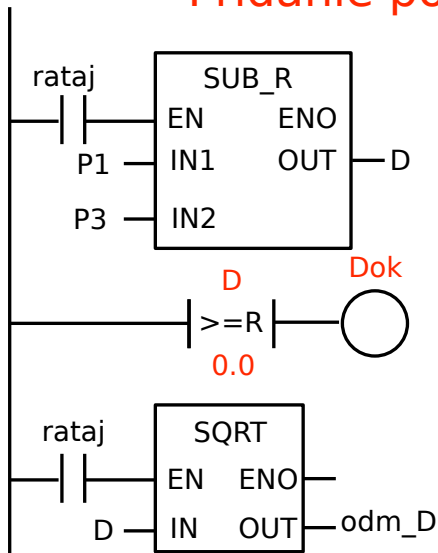
Príklady porovnávacích operácií



Konkrétny príklad

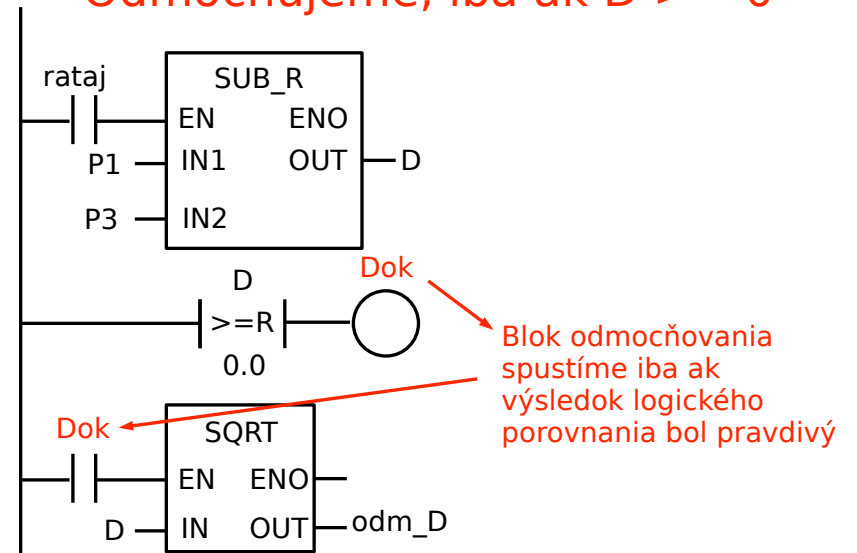


Pridanie porovnania



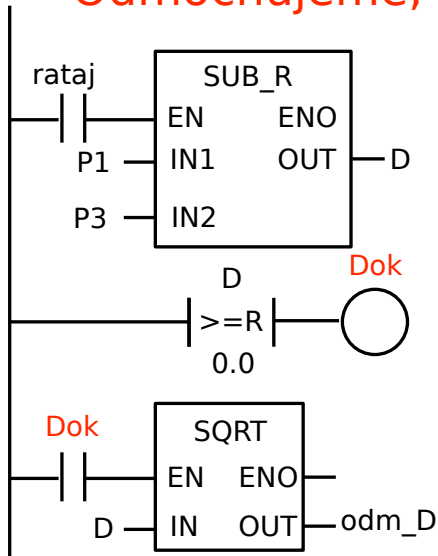
Platí, že " $\text{D} \geq 0.0$ " ?
 Ak áno, $\text{Dok} = 1$
 Ak nie, $\text{Dok} = 0$

Odmocňujeme, iba ak $\text{D} \geq 0$



Blok odmocňovania spustíme iba ak výsledok logického porovnania bol pravdivý

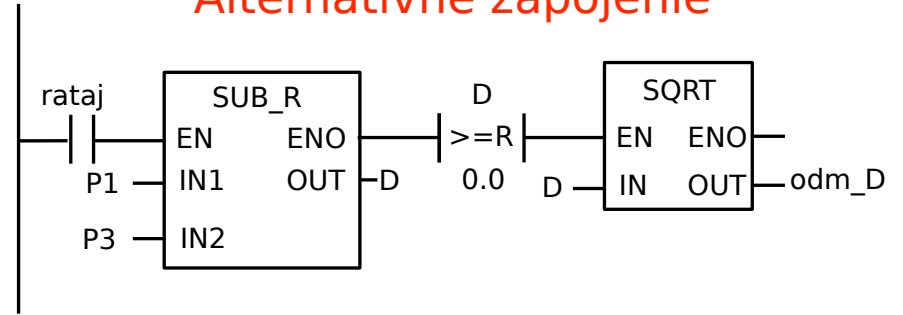
Odmocňujeme, iba ak $D \geq 0$



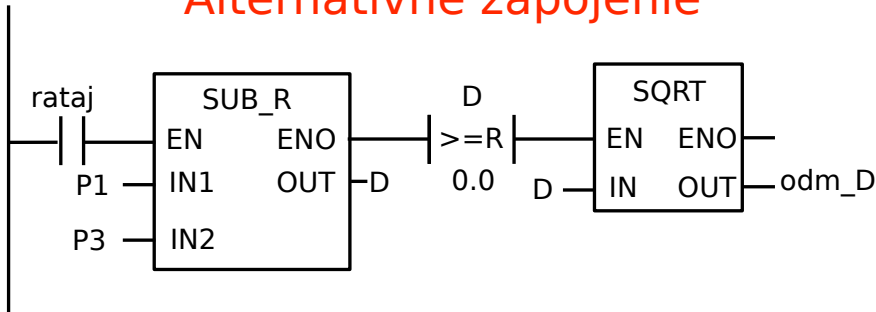
Nezabudnite premennej "Dok" priradiť pamäťovú adresu!!!

Kedže výsledkom porovnávania je iba 1 alebo 0, "Dok" bude bitová premenná, napr. s adresou Q0.0

Alternatívne zapojenie



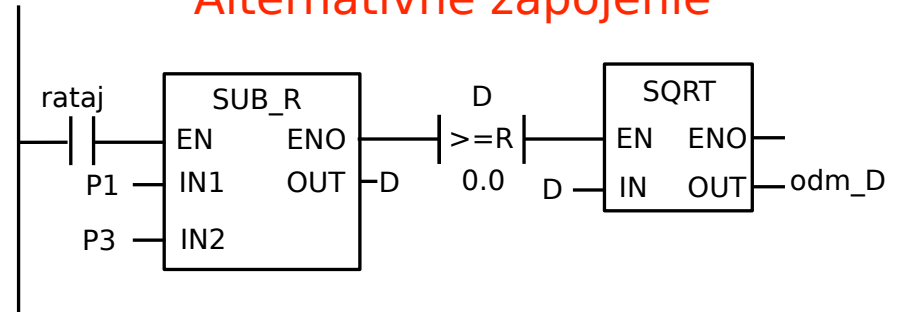
Alternatívne zapojenie



Výhoda: nepotrebujeme premennú Dok

Nevýhoda: nemáme v pamäti uložený výsledok porovnania a tým pádom ho nemôžeme využiť na spustenie iných operácií (museli by sme robiť porovnanie znova)

Alternatívne zapojenie



Výhoda: nepotrebujeme premennú Dok

Nevýhoda: nemáme v pamäti uložený výsledok porovnania a tým pádom ho nemôžeme využiť na spustenie iných operácií (museli by sme robiť porovnanie znova)

Zhrnutie

- Matematické bloky dokážu vykonať iba základné operácie medzi dvoma číslami
 - čiže $y = x^3$ musíme zapísať pomocou "z = x*x" a "y = z*x"
- Čísla môžu byť buď zadané "natvrdo" v programe, alebo uložené v pamäťových adresách (v tom prípade musíme do pamäte nahráť konkrétne číslo pomocou bloku MOV_R)
- Je potrebné vytvoriť pamäťové adresy aj pre dočasné výsledky ("z" v príklade hore)

Analýza matematického výrazu

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

- Koľko premenných budeme potrebovať?
- $P1 = b*b$, $P2 = 4*a$, $P3 = P2*c$, $D = P1 - P3$,
 $odm_D = \text{sqrt}(D)$, $P4 = 2*a$, $P5 = -b$,
 $P6 = P5 + odm_D$, $P7 = P5 - odm_D$,
 $x1 = P6/P4$, $x2 = P7/P4$
- Pre každú, aj pomocnú, premennú je potrebné vytvoriť pamäťovú adresu (VD0, VD4, VD8, ...)