RA   Tobias Brunner   ①
                 Olivier Cathrein

1.)   CPU 500 Mhz → 2nsec = 1 Taktzyklus   (V04, F.5)

ALU	4 nsec	2	2	2
LOAD	8 nsec	4	6	4
STORE	12 nsec	6	6	3
BRANCH	6 nsec	3	3	3
Durschschnitt:		3.75	4.25	3

2.) $(3.75/4.25) - 1 = 11.77\%$ langsamer

$(3.75/3) - 1 = 25\%$ schneller

b.)

ALU	50%	4 nsec 2	→ 1	
LOAD	15%	8 nsec 4	→ 0.6	
STORE	25%	12 nsec 6	→ 1.5	hier kann man Zeit gewinnen.
BRANCH	10%	6 nsec 3	→ 0.3	
			+	
			3.4	

RA

2) - max. 4 inputs, subroutine die mehr inputs braucht, braucht
Stack für die zusätzlichen.

- return Adresse wird auf den Stack gepushed damit sie nicht
verloren geht/überschrieben

3.) slt in ALU (set less than), VO5, 10-12
Steuerleitung: Anivert Bnegate Operation
slt :   0   1   11

- Bekommt den "Less value„ vom input raus.

$$A \text{ slt } B \begin{cases} 00...001 \text{ wenn } A-B<0 \\ 00...000 \text{ wenn } A-B\geq0 \end{cases}$$

4.) Die 32-Bit Konstante wird in 2 Schritten geladen

    – 16-Bits in obere 16-Bits von $s0

     lui $s0 ,"die linken 16-Bits„

    – danach die unteren 16-Bits einfügen.

     ori $s0, $s0 "die rechten 16-Bits„

5.) ALU: OPCodes ⟶ A,B not inverted, result is on the out of AND-Gate

AND	0000
OR	0001 ⟶A,B not inverted, result is on the out of OR-Gate
ADD	0010 ⟶A,B not inverted, use full adder to calculate (A+B)
SUBTRACT	0110 ⟶B is inverted, use full adder to calculate (a+(-b))=(a-b)
SLT	0111 ⟶Op 3 is chosen
NOR	1100 ⟶A,B inverted, Result is on out of AND-Gate

      A NOR B = 1(A or B) = 1A AND 1B