

Part1 için hazırlanan notlar:

Basamak Eşitleme fonksiyonu ayrıntısı;

Payda Basamak Sayısı (Payda1) \rightarrow ilkönce $S5$ kullanıldı, sonra $t1$ 'e
 " " (Payda2) $\rightarrow T2$

if ($t1 > t2$) $\rightarrow t4$

$t3 = \$S3$, $t0 = t2$

else

$t3 = \$t1$, $t0 = t1$

while ($t0 < S4$) \rightarrow basit için $t5$ settas

{
 $t3 \neq 10$
 $t0++$

}
 if ($t1 > t2$)

$S3 = t3$

else

$S1 = t3$

En büyük
 Basamak sayısı
 temsil
 ediyor.

Basamak Eşitle =

Verilen sayıların payda uzun-
 lüğünü eşit hale getirir.
 içerisinde $S4$ yeni max
 basamak uzunluğu kullanıldı.

Not = $\$S3$ ve $\$S1$ registerları Payda2 ve
 Payda1 değerlerini alabilmek için kullanılmıştır.

2 sayının Basamak Sayısının en büyük olanını bulan fonksiyon

int $t0 = 0$;

$t1 = \$a0$

$t2 = \$a1$

$t3$ is used for slt

$t4$ is used for slt

$t5$ is used for or

$t0$ is setted to

while ($t1 > 0$ || $t2 > 0$) $\rightarrow t5$

{
 $t1 /= 10$;

$t2 /= 10$;

$t0++$;

return $t0$;

Payda Basamak Sayısı

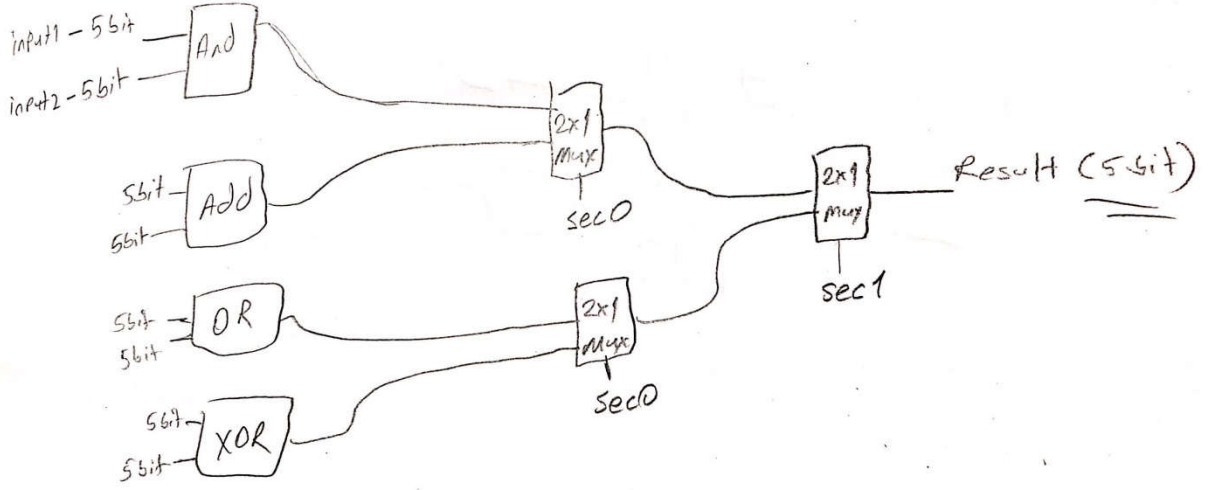
Pay ile paydayı birleştirip integer yapma fonksiyonu ve Toplama , çıkarma çarpma fonksiyonlarının algoritması;

$A_0 = \text{Pay}$, $A_1 = \text{Payda}$: Bu fonksiyon kullanılır.
 1. $t_0 = a_0$
 $t_7 = 0$ # counter.
 while ($t_7 < S_4$) condition içerdiği slt ile t_1 'e set edildi.
 { max kayanlık sayısı
 $t_0 * = 10;$
 t_7++
 }
 if ($a_0 > 0$)
 add $\$t_0, \$t_0, \$a_1$
 else
 sub $\$t_0, \$t_0, \$a_1$
 return t_0
Toplama - Çıkarma - Çarpma
 Sayıları birleştirir (S_0, S_1) $\rightarrow S_5$
 Sayıları birleştirir (S_2, S_3) $\rightarrow t_1$
 $t_0 = S_5$
 $\$t_2 = \$t_0 + \$t_1 \rightarrow$ Çarpma ve çıkarma
 $\$t_3 = 0, \$t_4 = 1$ işlemleri için burası
 değişir.
 while ($t_3 < S_4$)
 { $t_4 = t_4 \times 10$
 t_3++ }

$\rightarrow t_5 = t_0 / t_1$
 $t_6 = t_2 / t_4$
 if ($t_5 < 0$)
 $t_5 * = -1$
 $v_0 = t_6$
 $S_4 = t_5$
 return;

Part2 için hazırlanan Notlar

likeALU Tasarımı;



Full Adder Tasarımı;

Full Adder

