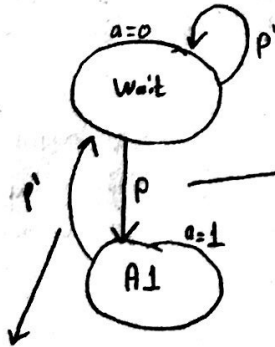


Projede öncelikle 2 tane taraf butonu kullandım. Fakat bu tuşları kullanan kişiler kullanırken olacağı için bir D flip-flop kullanarak metastability oluşturma riskini yüksek oranda azaltmış oldum. Ardından buton senkronizer kullanarak bir kişinin 1 tuş 1 cycle'dan fazla basınca sürekli basıyormuş gibi olmasını engelledim. Burdan çıkan outputlarla ve devrenin anlık state'i ile next state'e karar verdim. Tüm devrenin outputları inputa bağlı değil, sadece anlık state'e bağlı olduğu için ledleri registerdan gelen s2, s1, s0 singalleriyle yaptım. Ayrıca Reset butonunu oyunun registerına ayrı olarak bağladım herhangi bir zamanda basınca ilk duruma geçirebildim. Logisim'de registerın içine ulaşamadığım için 000 state'ini başlangıç durumu yaptım böylelikle reset tuşu tüm flip-flopı 0'ladığı için ilk duruma dönebildim.

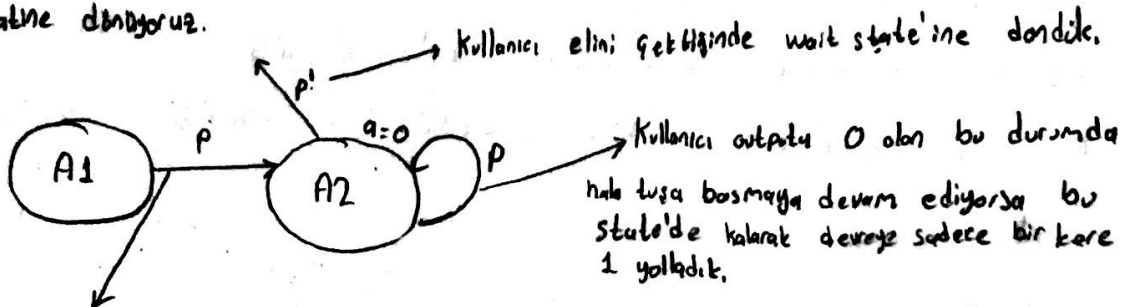
### System Details

Buton Synchronizer için 3 state belirledim. İlk wait state'i eğer butona basılmadıysa output'u 0 olan bu state'de kalarak devreye yolladığımız input her positive edge'de 0 olarak kalır.



Eğer butona basılırsa output'u 1 olan ve oyun devresine 1 inputunu yollayan A1 state'ine geçtik.

Eğer butona basıldıktan sonraki positive edge geldiğinde kullanıcı elini kaldırırsa wait state'ine döndürürüz.



Kullanıcı elini çektiğinde wait state'ine döndük.

Kullanıcı output'u 0 olan bu durumda hala tuşa basmaya devam ediyorsa bu state'de kalarak devreye sadece bir kere 1 yolladık.

Eğer kullanıcı A1 state'ine geldiğinde hala tuşa basıyorsa, output'u 0 olan

A2 state'ine geçtik.

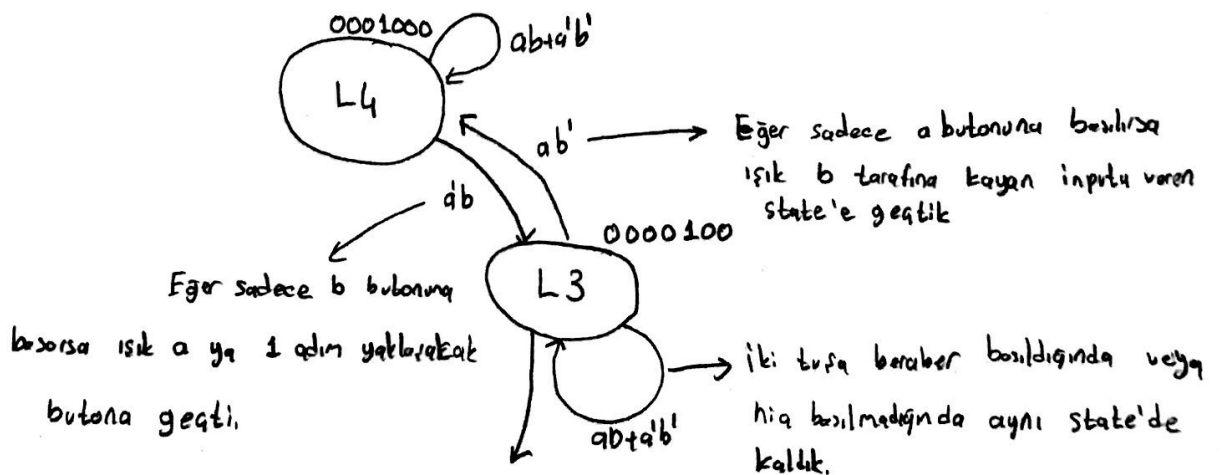
Böylelikle kullandığımız 3 state oluştuk. 3 state kullanmamız,  $(3 \leq 2^2) \rightarrow 2$  bit register kullanarak bu sorunu çözeceğimizi anlamına geldiği için 2 bitlik register kullanarak bu problemi çözdüm. 2 buton için de bu fsm'yi kullandım.

Butonlardan gelen inputları oyunumuz için kullandığım ana fsm'ye yolladım. 7 farklı led durumu olduğu için 7 state'den oluşan bu fsm'de  $(7 \leq 2^3) \rightarrow 3$  bitlik state register kullanarak makinemizi tasarlayabileceğimizi anlamına geldiği için 3 bitlik register kullanıp devreye 5 input sağlamış olduk. 5 input, 3 next state ve 7 led çıkışı olduğu için toplamda 10 outputumuz oldu.

	$S_2$	$S_1$	$S_0$	a	b	$n_2$	$n_1$	$n_0$
$L_4$	0	0	0	0	0	0	0	0
	0	0	0	0	1	1	0	0
	0	0	0	1	0	0	0	1
	0	0	0	1	1	0	0	0
$L_3$	0	0	1	0	0	0	0	1
	0	0	1	0	1	0	0	0

$S_2$	$S_1$	$S_0$	$a_7$	$a_6$	$a_5$	$a_4$	$a_3$	$a_2$	$a_1$
0	0	0	0	0	0	1	0	0	0
0	0	1	0	0	0	0	1	0	0
0	1	0	0	0	0	0	0	1	0

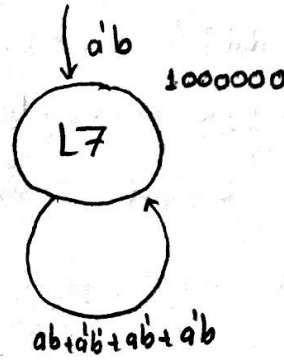
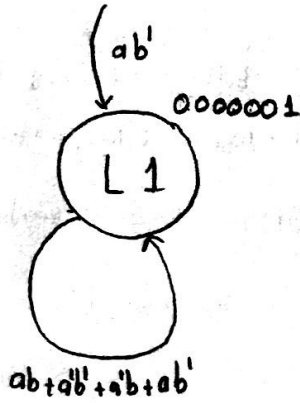
Şeklinde 2 truth table oluşturdum çünkü sağ tablodan görüldüğü üzere outputlar onluk a ve b inputlarına bağlı değil. a ve b inputları sadece devrenin next state'i için etkili.



San state'ler (oyunu bitirme durumları) heriç her state aynı mantıkla çalıyor.

Butonlara aynı anda basılma ve hiç basılma durumu, onlik state'i değitirmediği için

bulduğumuz state'lerde kaldım. Son state'lerde oyun bittiği için sadece o state'e geliş durumu var.



\* Ara state'ler önceki state'ler ile aynı mantığa sahip.

- Devre de asenkronluğu engellemek için kullanıln 1 cycle gecikme yapıyor.
- Buton Synchronizer'lar da 1 cycle gecikme yapıyor,

Toplamda ana fsm'ye 2cycle gecikmeyle hata oranı çok düşük inputlar yollamış bulundum.

Tüm devre aynı clocktan beslendiği için tüm devre senkronize şekilde inputları işlemiş oldu.

## Test

- TuŖlara basılmama durumu: Devrede hiĖbir tuŖa basılmadığında hangi durumdaysa sŖrekli o durumda kaldı.
- İki tuŖa basılma durumu: Devrede herhangi bir durumda iki tuŖa beraber basıldığında durum deĖiŖmedi.
- Sadece bir tuŖa basılma durumu: Devrede a veya b tuŖlarından sadece birine basıldığında led karsı tarafa 1 adım yaklaştı.
- Basılı tutma durumu: Herhangi bir tuŖa basılı tutulunca sadece 1 kere ilerledi, basılı tutulan bir tuŖ varken diĖer tuŖa basıldığında sadece diĖer tuŖa basılmış gibi ilerledi.
- Reset durumu: Herhangi bir anda resete basıldığında oyun baŖlangıĖ konumuna dördü.