

**Gebze Technical University
Computer Engineering**

CSE 433 - 2020 Spring

Project 1 REPORT

**EMİRHAN KARAGÖZOĞLU
151044052**

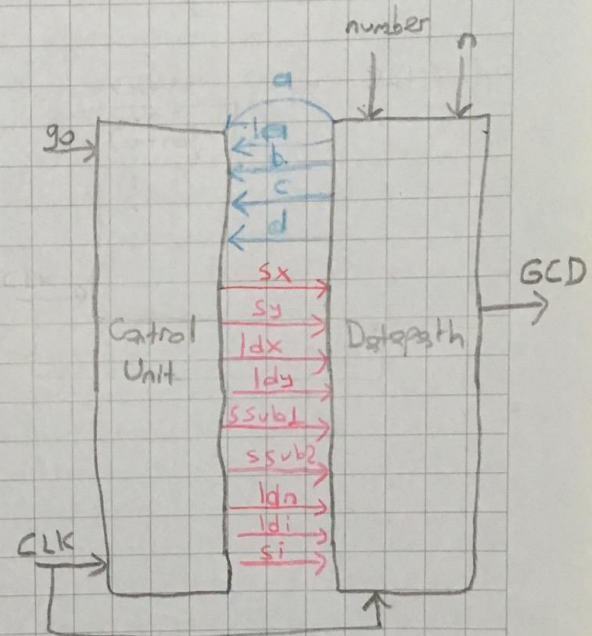
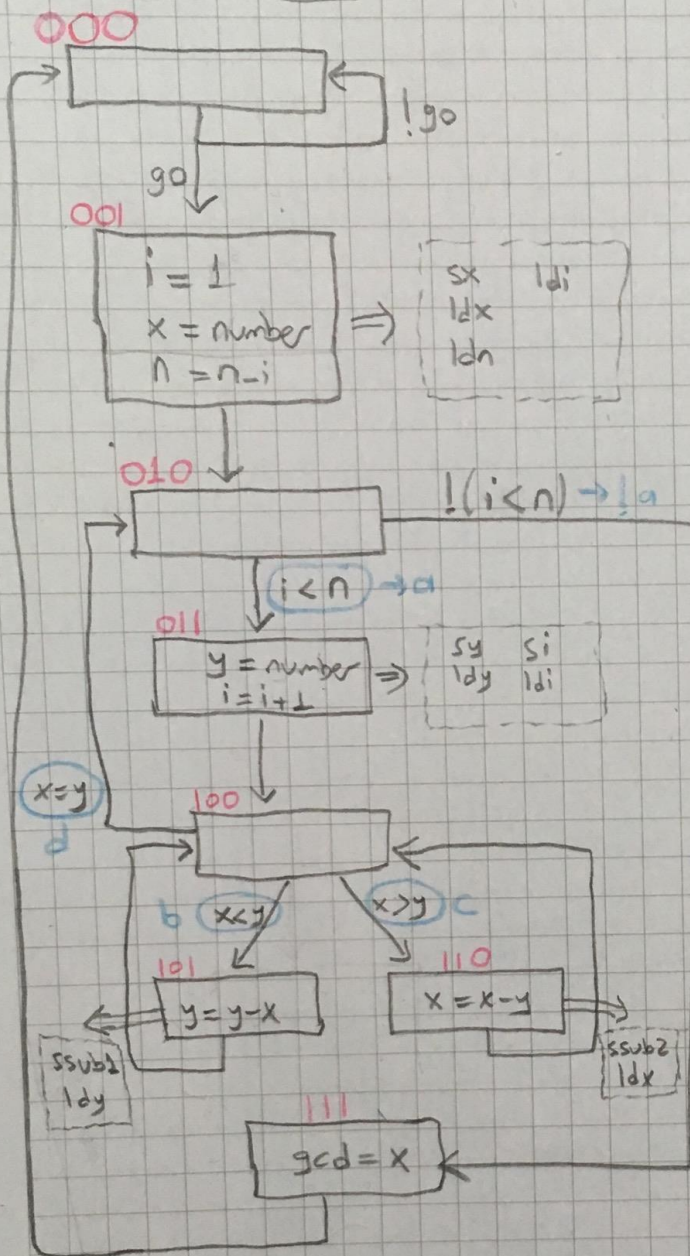
Course Assistant: Fatma Nur Esirci

C Code

```
1 void gcd(int arr[],int n_i,int go){
2     int x,y,i,n,gcd;
3
4     while(1){
5         while (!go);
6         i=1;
7         x=arr[i]; //number
8         n=n_i;
9
10        while(i<n){
11
12            y=arr[i]; //number
13            ++i;
14
15            while(x!=y){
16                if(x<y)
17                    y=y-x;
18                else
19                    x=x-y;
20            }
21
22        }
23        gcd = x;
24    }
25 }
```

State Diagram

State diagram



State Transition Table and Boolean Expressions

Present			Inputs							Next		
P2	P1	P0	go	!a	a	b	c	d		N2	N1	N0
0	0	0	0	x	x	x	x	x		0	0	0
0	0	0	1	x	x	x	x	x		0	0	1
0	0	1	x	x	x	x	x	x		0	1	0
0	1	0	x	1	0	x	x	x		1	1	1
0	1	0	x	0	1	x	x	x		0	1	1
0	1	1	x	x	x	x	x	x		1	0	0
1	0	0	x	x	x	1	0	0		1	0	1
1	0	0	x	x	x	0	1	0		1	1	0
1	0	0	x	x	x	0	0	1		0	1	0
1	0	1	x	x	x	x	x	x		1	0	0
1	1	0	x	x	x	x	x	x		1	0	0
1	1	1	x	x	x	x	x	x		0	0	0

$$N2 = (P2.P1'.P0) + (P2.P1.P0') + (P2'.P1.P0) + (P2.P1'.P0'.(b+c)) + (P2'.P1.P0'.!a)$$

$$N1 = (P2'.P1.P0') + (P2'.P1'.P0) + (P2.P1'.P0'.(c+d))$$

$$N0 = (P2'.P1.P0') + (P2'.P1'.P0'.go) + (P2.P1'.P0'.b)$$

$$\rightarrow S5 + S6 + S3 + (S4.(b+c)) + (S2.!a)$$

$$\rightarrow S2 + S1 + (S4.(c+d))$$

$$\rightarrow S2 + S0.go + S4.b$$

$$sx = S1$$

$$sy = S3$$

$$si = S3$$

$$ssub1 = S5$$

$$ssub2 = S6$$

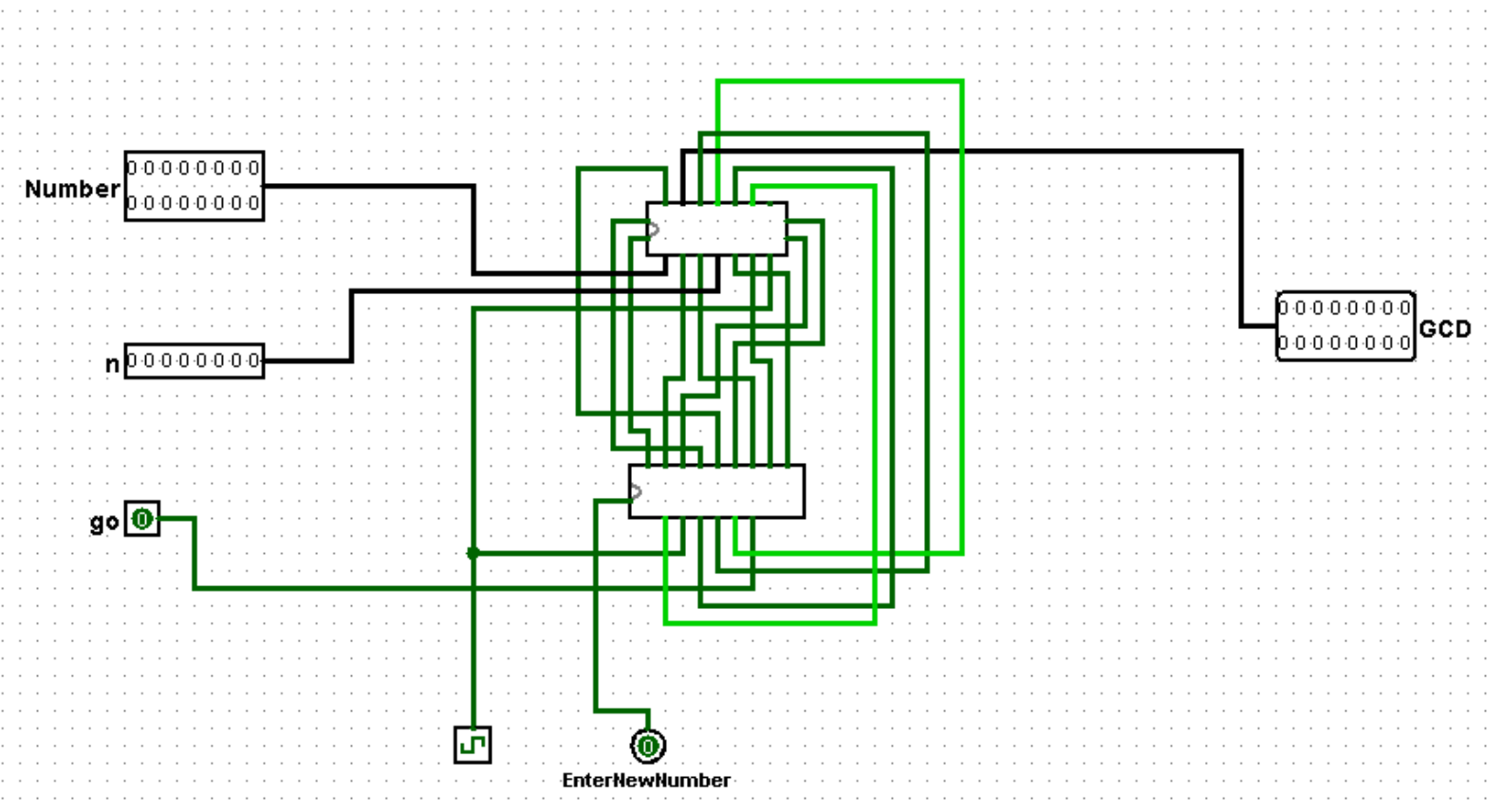
$$ldx = S1+S6$$

$$ldy = S3+S5$$

$$ldn = S1$$

$$ldi = S1+S3$$

Testing



Logisim 2.7.1 versiyonu ile .circ uzantılı GCD dosyasını açıyoruz. Bu devre içerisinde datapath (üstteki kutu) ve control unit (alttaki kutu) Logisim library olarak kullanılıp bağlantıları ayarlanmıştır. Adım adım uygulanması gereken test süreci aşağıdaki gibidir;

1. GCD'si hesaplanmak istenen sayılardan ilki number inputuna girilir.
2. Kaç tane sayının GCD'si hesaplanıyorsa n inputuna bu değer girilir.
3. go sinyali 1 yapılır.
4. Clock bir kez tetiklenir ve go sinyali 0 yapılır.
5. Bu aşamadan sonra her clock tetiklendiğinde "EnterNewNumber" outputu takip edilmeli ve bu değer 1 olduğu zaman sırada yeni number girilip clock ondan sonra tetiklenmeye devam edilmeli.
6. İşlemler bittikten sonra GCD outputunda girilen sayıların gcd değeri gözlenebilir.

Çalışma prensibi olarak girilen 2 sayının gcd'si hesaplanıyor ve bu değer ilk sayı olan x register'ına kaydediliyor. Ardında sıradaki sayı y registerına okunuyor ve bu işlem sayılar tamamlanana kadar devam ediyor. Örneğin;

GCD(48,8,16,4)

1.step-> gcd(x=48,y=8) = 8

2.step->gcd(x=8,y=16) = 8

3.step->gcd(x=8,y=4) = 4 -> gcd = x =4 Done.

Herhangi iki sayının gcd'sini hesaplandığı süreçte (içerdeki while döngüsü) yeni bir number girilirse y registerı o anda girilen değeri içerisine yazamaz. Ancak o iki sayının gcd'si hesaplandıktan sonra içerdeki döngüden çıkar ve sıradaki sayıyı okuması için control unitten gerekli sinyaller geldiği zaman yeni gelen sayıyı okuyabilir. Bu süreç numberları doğru zamanda gelmesi için "EnterNewNumber" output eklenerek görsel kolaylık sağlanması hedeflendi. Bu sinyal 1 olmadan gelen numberın değiştirilmesi anormal sonuçlar ortaya çıkartabilir.

Her clock tetiklemesi yapıldığında datapath ve control unit içerisine girilip sinyaller, present stateler, register değerleri gözlenebilir ve state diagramdaki değerler ile karşılaştırılıp doğruluğu kontrol edilebilir. İşlem tamamlandıktan sonra başlangıç state'ine döndüğü için 1. adımdan başlayarak tekrar test işlemi uygulanabilir. Test işlemi esnasında herhangi bir nedenden testi sonlandırıp başa dönmek için Simulate->Reset Simulation yapılmalıdır.