

biology - Virus

Organisms & non-living things are agents of
diseases. Non-living things can't cause disease
but living things can. Living things can cause
diseases. Viruses are very small. They can't
live without host cells. They can't move on their
own. They can only multiply inside the host cells.
They can't live outside the host cells.

Chemistry & Physics - Rope

Earth Science - Climate change

لایه مصلح در اینجا به فرود خود را که باعث شد که CO_2 و گازهای دیگر را از زمین بخواهد. در این میان پس از بازگشت این گازها به زمین، نیز آنها به زمین بازگردند. در این میان پس از بازگشت این گازها به زمین، نیز آنها به زمین بازگردند. در این میان پس از بازگشت این گازها به زمین، نیز آنها به زمین بازگردند.

Social Science - Rumor Mill

agent یعنی افراد را در یک شبکه می‌دانیم که در آنها این اتفاقات رخ داده باشند. این اتفاقات می‌توانند از این دو نوع باشند:
۱) اتفاقاتی که در یک زمان محدود رویداد می‌افتد و پس از آن اتفاقاتی که در طول زمان می‌افتد.
۲) اتفاقاتی که در یک زمان محدود رویداد می‌افتد و پس از آن اتفاقاتی که در طول زمان می‌افتد.
۳) اتفاقاتی که در یک زمان محدود رویداد می‌افتد و پس از آن اتفاقاتی که در طول زمان می‌افتد.
۴) اتفاقاتی که در یک زمان محدود رویداد می‌افتد و پس از آن اتفاقاتی که در طول زمان می‌افتد.
۵) اتفاقاتی که در یک زمان محدود رویداد می‌افتد و پس از آن اتفاقاتی که در طول زمان می‌افتد.
۶) اتفاقاتی که در یک زمان محدود رویداد می‌افتد و پس از آن اتفاقاتی که در طول زمان می‌افتد.
۷) اتفاقاتی که در یک زمان محدود رویداد می‌افتد و پس از آن اتفاقاتی که در طول زمان می‌افتد.
۸) اتفاقاتی که در یک زمان محدود رویداد می‌افتد و پس از آن اتفاقاتی که در طول زمان می‌افتد.
۹) اتفاقاتی که در یک زمان محدود رویداد می‌افتد و پس از آن اتفاقاتی که در طول زمان می‌افتد.
۱۰) اتفاقاتی که در یک زمان محدود رویداد می‌افتد و پس از آن اتفاقاتی که در طول زمان می‌افتد.

Behavioral

```
Sum(in1, in2, out){  
    out ← in1 + in2 after 2ns  
}  
    }  
    Minus (in1, in2, out) {  
        out ← in1 - in2 after 1ns  
    }  
    }  


---

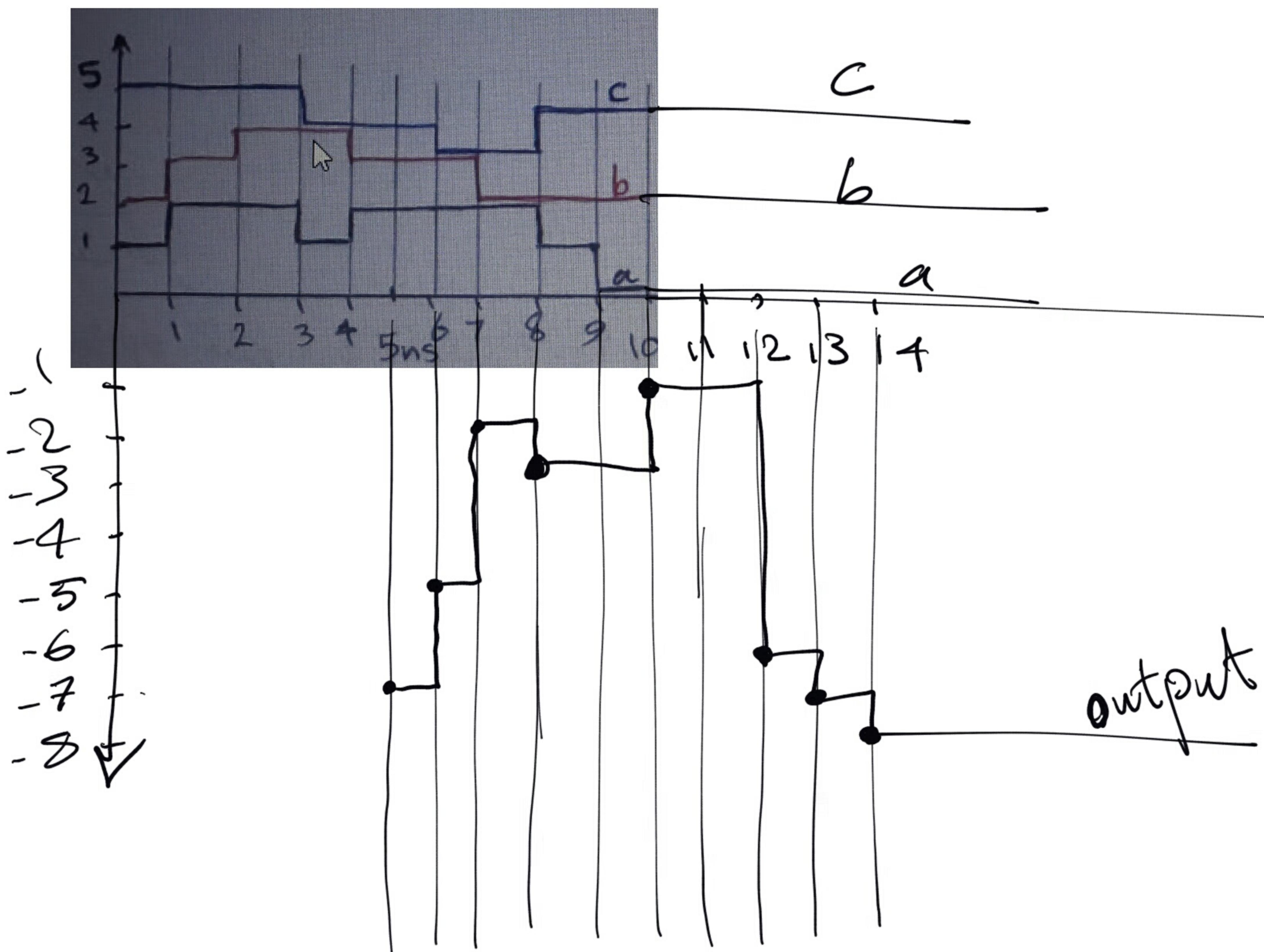
  
MultiplyTwo (in, out){  
    out ← in * 2 after 3ns  
}
```

Structural

```
Sys(in1,in2,in3,out){  
    Map(Sum,in1,in1)  
    Map(Sum,in2,in2)  
    Map(MultiplyTwo,in3,in3)  
    Map(Minus,in1,Sum.out)  
    Map(Minus,in2,MultiplyTwo.out)  
    Map(out,Minus.out)  
}
```

Transaction Queue

0	a	(0,1) (1,2) (3,1) (4,2) (8,1) (9,0)
2	b	(0,2) (1,3) (2,4) (4,3) (7,2)
5	c	(0,5) (3,4) (6,3) (8,5)
2	+out	(2,3) (3,5) (4,6) (5,5) (9,4) (10,3) (11,2)
10	*2out	(1,10) (4,8) (7,6) (9,10)
	-out outpt	(5,-7) (6,-5) (7,-2) (8,-3) (10,-1)
	deck	(12,-6) (13,-7) (14,-8)



Codjil

Data Flow

begin

$$\text{output} \leftarrow ((a + b) - (c * 2))$$

end