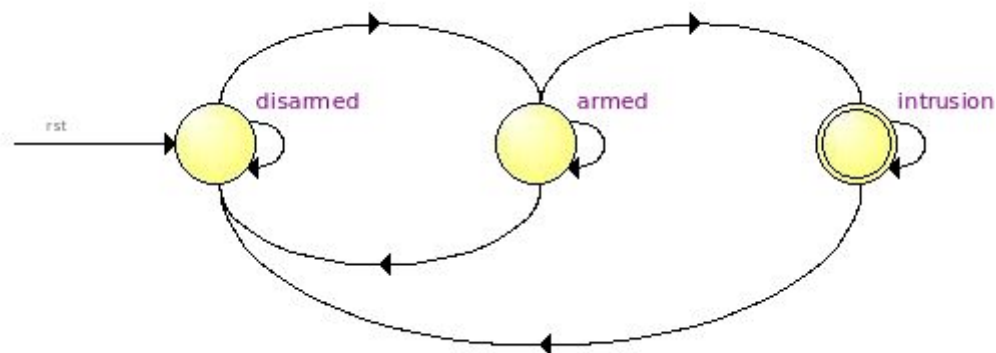


Exercício: Faça o download do arquivo `simple_car_alarm.qar`, disponível no Moodle da disciplina e realize as seguintes avaliações:

1. Perceba que existem 4 versões para o mesmo Alarme de Carro. Realize a simulação no ModelSim das 4 versões e perceba qual é a diferença de funcionamento desses circuitos.
2. Anote o número de elementos em cada versão.
3. Qual versão é a mais adequada na sua opinião?
4. Modifique a sua escolha para que o alarme não possa ser ativado se o "sensor" estiver em "1".
5. Analise o arquivo `tb_vX.do` e modifique-o para testar também essa nova condição.
6. Implemente a nova versão como uma nova arquitetura "fsm_v5", e escrever o arquivo de simulação "tbv5.do".
7. Salve as telas da simulação ("v5_sim.png"), tela da fsm ("v5_fsm.png"), tela do RTL ("v5_rtl.png").
8. Acrescente os novos arquivos no projeto e salve o novo .qar

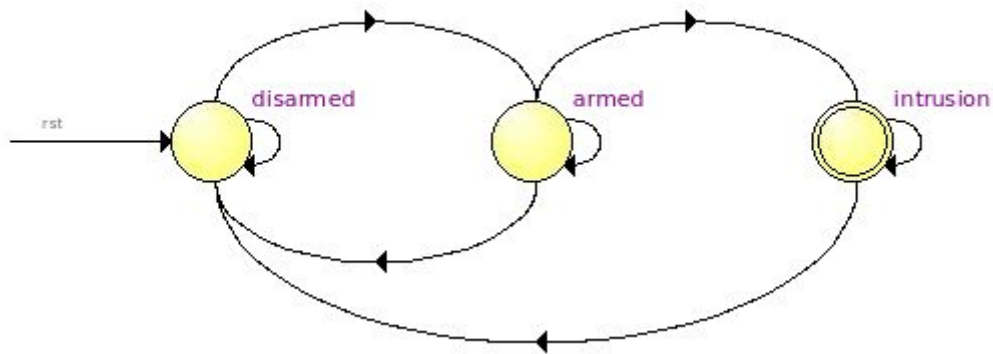
QUESTÃO 01

Versão 01: 3 estados (*running*)



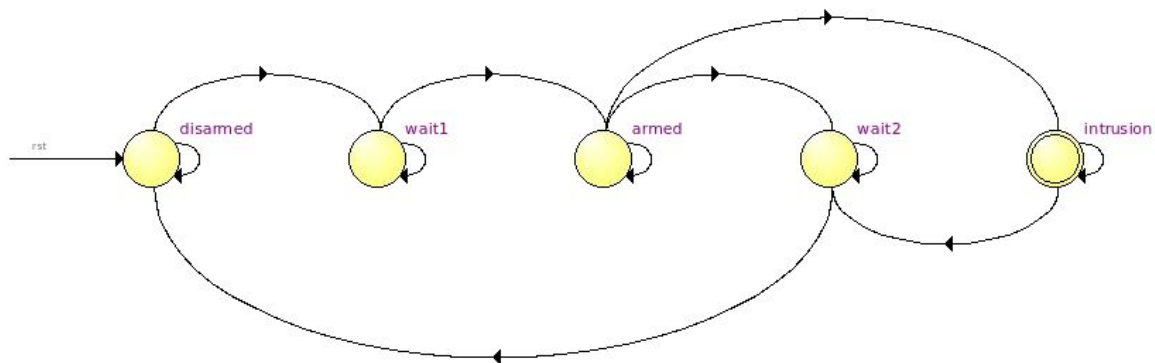
	Source State	Destination State	Condition
1	armed	intrusion	(sensors)
2	armed	disarmed	(remote).(!sensors)
3	armed	armed	(!remote).(!sensors)
4	disarmed	disarmed	(!remote)
5	disarmed	armed	(remote)
6	intrusion	intrusion	(!remote)
7	intrusion	disarmed	(remote)

Versão 02: 3 estados (*flag*)



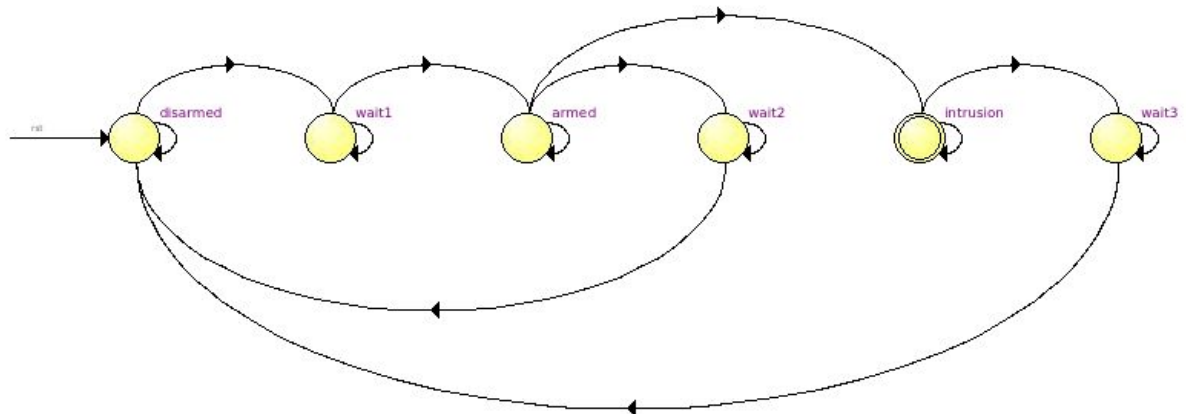
	Source State	Destination State	Condition
1	armed	intrusion	(sensors)
2	armed	disarmed	(process_2).(!sensors)
3	armed	armed	(!process_2).(!sensors)
4	disarmed	disarmed	(!process_2)
5	disarmed	armed	(process_2)
6	intrusion	intrusion	(!process_2)
7	intrusion	disarmed	(process_2)

Versão 03: 5 estados (2 *wait*)



	Source State	Destination State	Condition
1	armed	armed	(!remote).(!sensors)
2	armed	intrusion	(sensors)
3	armed	wait2	(remote).(!sensors)
4	disarmed	wait1	(remote)
5	disarmed	disarmed	(!remote)
6	intrusion	intrusion	(!remote)
7	intrusion	wait2	(remote)
8	wait1	armed	(!remote)
9	wait1	wait1	(remote)
10	wait2	wait2	(remote)
11	wait2	disarmed	(!remote)

Versão 04: 6 estados (3 wait)



	Source State	Destination State	Condition
1	armed	wait2	(remote).(!sensors)
2	armed	intrusion	(sensors)
3	armed	armed	(!remote).(!sensors)
4	disarmed	wait1	(remote)
5	disarmed	disarmed	(!remote)
6	intrusion	wait3	(remote)
7	intrusion	intrusion	(!remote)
8	wait1	wait1	(remote)
9	wait1	armed	(!remote)
10	wait2	wait2	(remote)
11	wait2	disarmed	(!remote)
12	wait3	wait3	(remote)
13	wait3	disarmed	(!remote)

QUESTÃO 02

Versão 01: 3 elementos lógicos

Flow Status	Successful - Fri Nov 8 14:03:52 2019
Quartus II 32-bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Full Version
Revision Name	simple_car_alarm
Top-level Entity Name	simple_car_alarm
Family	Cyclone
Device	EP1C3T100A8
Timing Models	Final
Total logic elements	3 / 2,910 (< 1 %)
Total pins	5 / 65 (8 %)
Total virtual pins	0
Total memory bits	0 / 59,904 (0 %)
Total PLLs	0 / 1 (0 %)

Versão 02: 4 elementos lógicos

Flow Status	Successful - Fri Nov 8 14:10:16 2019
Quartus II 32-bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Full Version
Revision Name	simple_car_alarm
Top-level Entity Name	simple_car_alarm
Family	Cyclone
Device	EP1C3T100A8
Timing Models	Final
Total logic elements	4 / 2,910 (< 1 %)
Total pins	5 / 65 (8 %)
Total virtual pins	0
Total memory bits	0 / 59,904 (0 %)
Total PLLs	0 / 1 (0 %)

Versão 03: 6 elementos lógicos

Flow Status	Successful - Fri Nov 8 14:12:06 2019
Quartus II 32-bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Full Version
Revision Name	simple_car_alarm
Top-level Entity Name	simple_car_alarm
Family	Cyclone
Device	EP1C3T100A8
Timing Models	Final
Total logic elements	6 / 2,910 (< 1 %)
Total pins	5 / 65 (8 %)
Total virtual pins	0
Total memory bits	0 / 59,904 (0 %)
Total PLLs	0 / 1 (0 %)

Versão 04: 4 elementos lógicos

Flow Status	Successful - Fri Nov 8 14:14:36 2019
Quartus II 32-bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Full Version
Revision Name	simple_car_alarm
Top-level Entity Name	simple_car_alarm
Family	Cyclone
Device	EP1C3T100A8
Timing Models	Final
Total logic elements	4 / 2,910 (< 1 %)
Total pins	5 / 65 (8 %)
Total virtual pins	0
Total memory bits	0 / 59,904 (0 %)
Total PLLs	0 / 1 (0 %)

QUESTÃO 03

As versões 2 e 4 possuem o menor número de elementos lógicos e evitam o *running*, mas entre as duas versões acredito que a mais adequada seria a versão 4 por que o mapa de estados é mais intuitivo, na versão 2 existe um elemento (*flag*) que não é considerado um estado e por isso não é exibido no mapa.

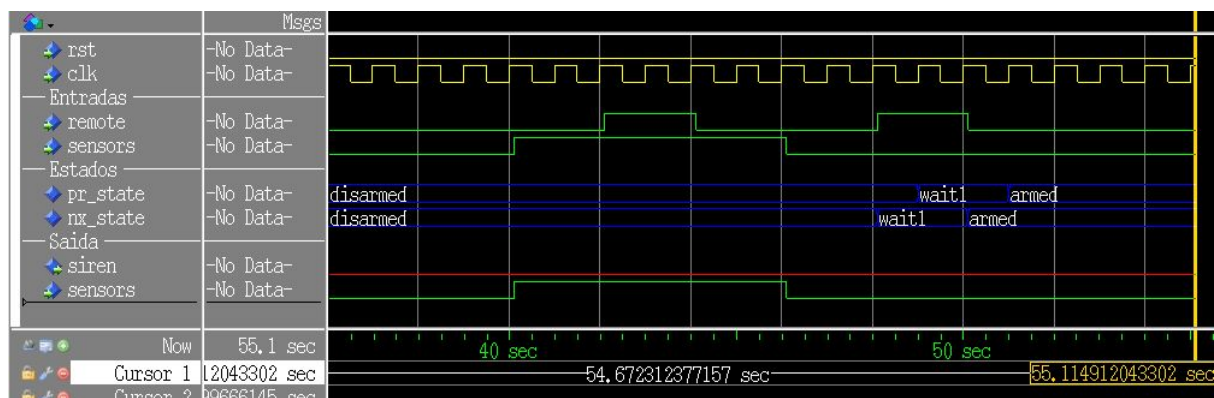
QUESTÃO 04

Modificação no código da versão 4 para versão 5:

```
when disarmed =>
    if (remote = '1' and sensors = '0') then
        nx_state <= wait1;
    else
        nx_state <= disarmed;
    end if;
```

QUESTÃO 05

Simulação: tentativa de armar o alarme com o sensor detectando presença



```
## linhas adicionadas no arquivo: tbv5.do
force -freeze sim:/simple_car_alarm/sensors 1 0, 0 6sec
force -freeze sim:/simple_car_alarm/remote 0 0, 1 2sec, 0
4sec, 1 8sec, 0 10sec
run 15 sec
WaveRestoreZoom {0 ps} {55sec}
```

QUESTÃO 06

Arquivo (.vhd):

https://github.com/camillabarreto/Codigos-VHDL/blob/master/AE4/simple_car_alarm_restored/simple_car_alarm.vhd

Arquivo (.do):

https://github.com/camillabarreto/Codigos-VHDL/blob/master/AE4/simple_car_alarm_restored/simulation/modelsim/tbv5.do

QUESTÃO 07

Imagens: <https://github.com/camillabarreto/Codigos-VHDL/tree/master/AE4/Imagens>

QUESTÃO 08

Arquivo (.qar): <https://github.com/camillabarreto/Codigos-VHDL/tree/master/AE4>