Sistemas Embarcados

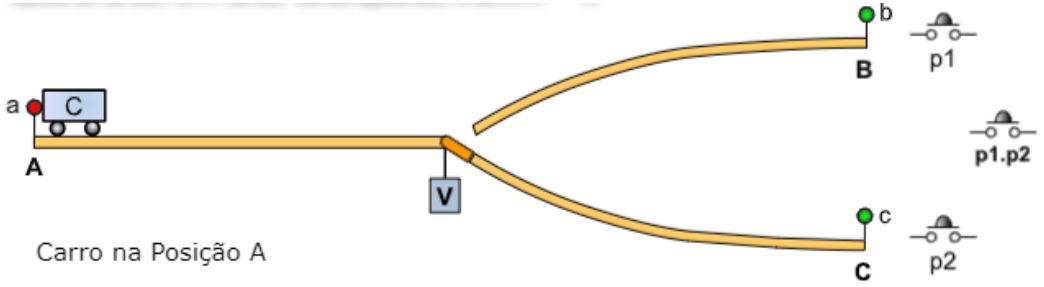
PROF. JOSENALDE OLIVEIRA

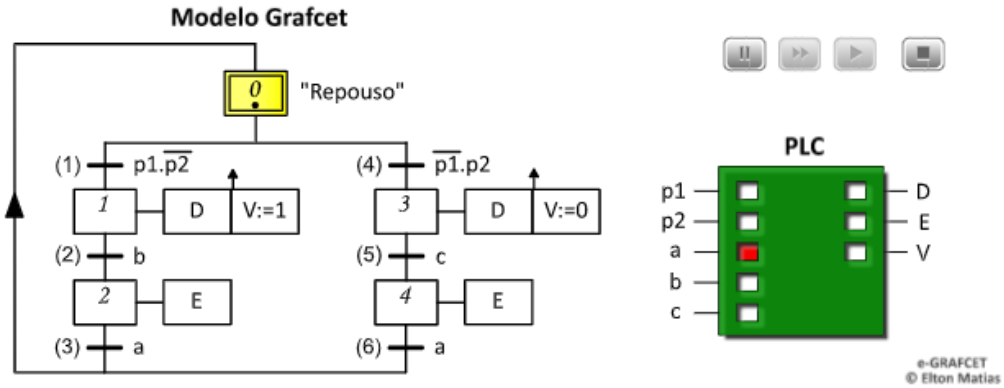
TADS UFRN

josenalde@eaj.ufrn.br
https://github.com/josenalde/embeddedsystems

Modelagem de processos sequenciais: GRAFCET

- Método gráfico para projetos de sistemas automatizados sequenciais (sugestão de vídeo aula: GRAFCET - Parte I (Em Português) – YouTube)
- Embora existam CLPs que permitam programar diretamente em GRAFCET (EN 60848), é mais conhecido como ferramenta de projeto, e o LADDER como uma linguagem de programação (portanto existem métodos de conversão GRAFCET-LADDER)
- Em computação, seria equivalente à análise do sistema, documentar solução





"**GR**Aphe **F**onctionnel de **C**ommande **E**tapes/**T**ransitions."

Ideia: A partir de etapas/condições iniciais, ocorre transição de etapas, onde em cada etapa uma ou mais ações são efetuadas. As transições possuem regras de disparo (lógica, temporização etc.)

Numa etapa pode-se ter um vetor de saídas (com seus estados 0, 1)

Relembrando...

- Máquina de estado
- Tabela de transições de estado

Table 3.2 Six states of TLC controller

State	*North–South LEDs*	*East–West LEDs*	*Delay in seconds*
0	Green	Red	15
1	Yellow	Red	3
2	Red	Red	3
3	Red	Green	15
4	Red	Yellow	3
5	Red	Red	3

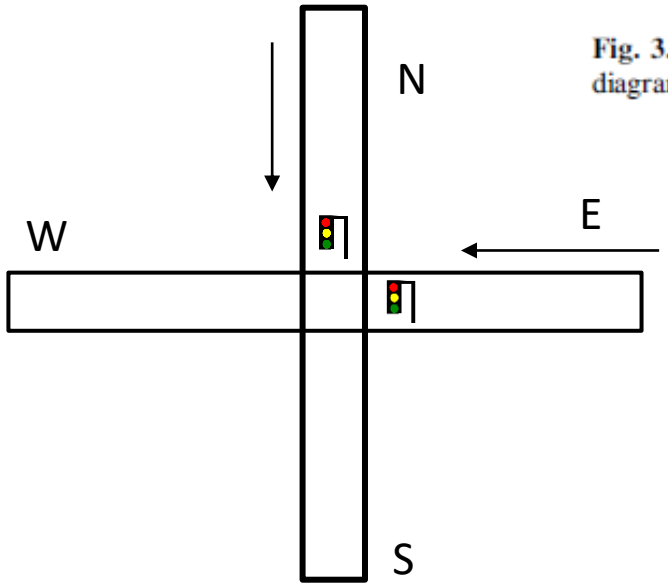
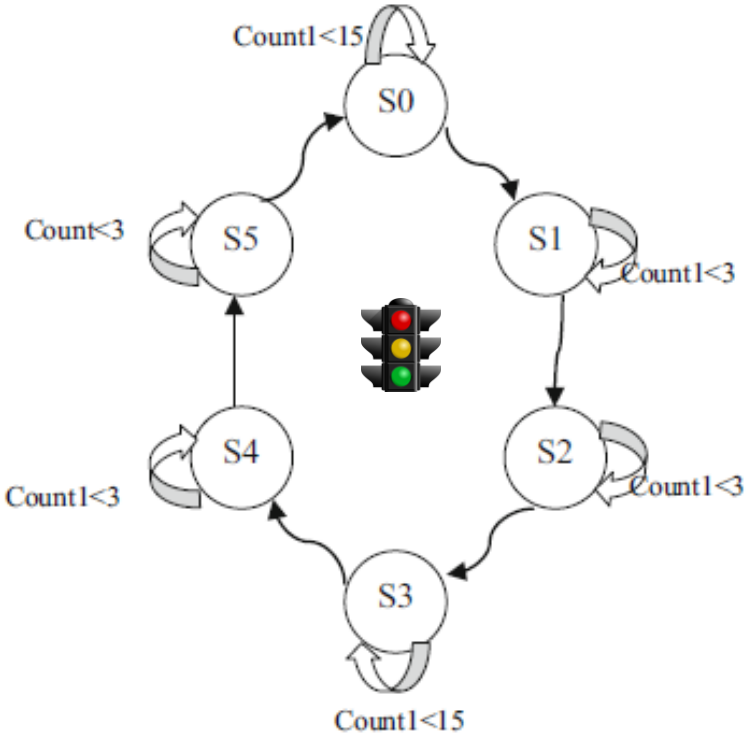
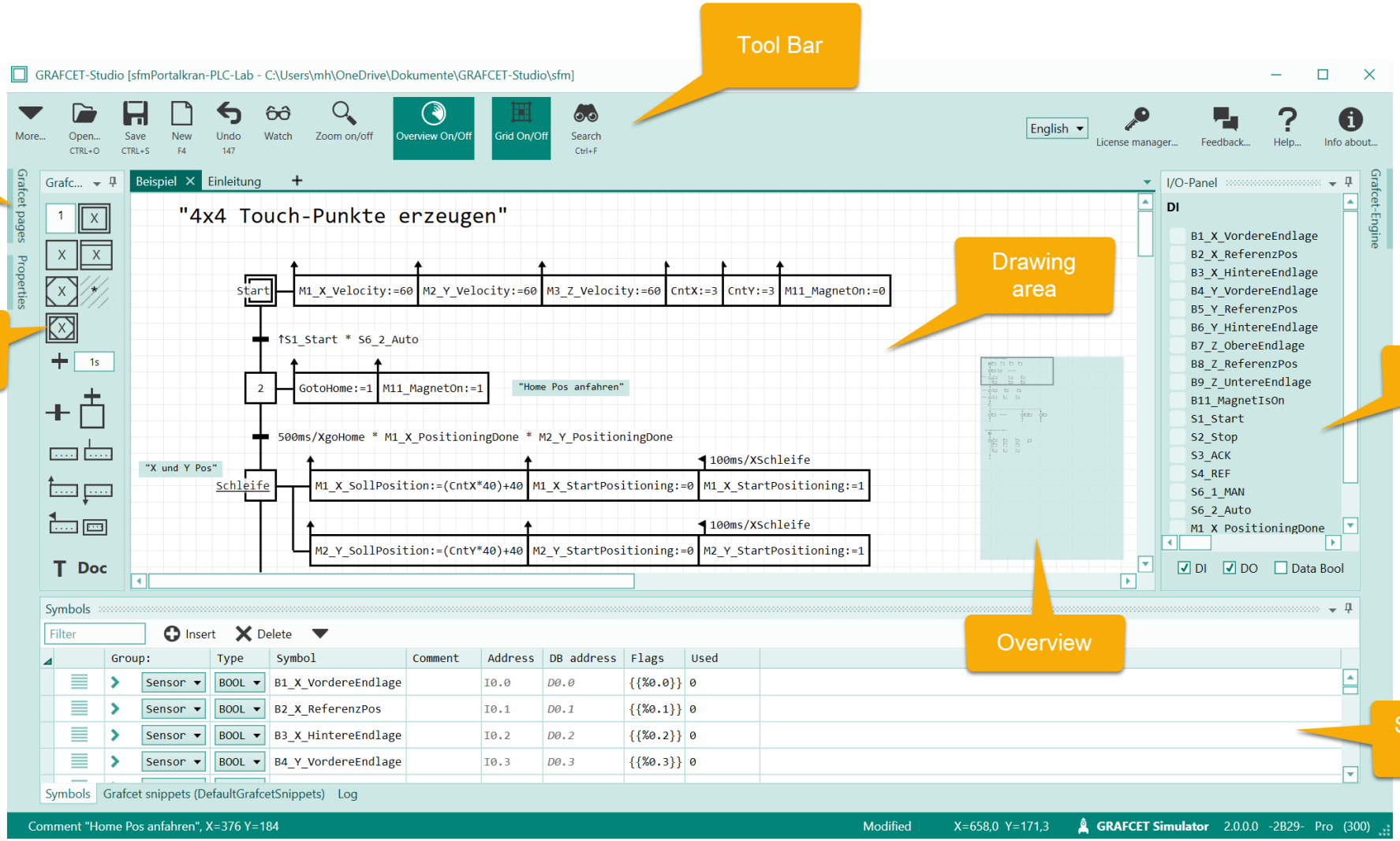


Fig. 3.21 State machine diagram



https://github.com/josenalde/embeddedsystems/blob/master/src/traffic_light.vhd

Modelagem de processos sequenciais: GRAFCET

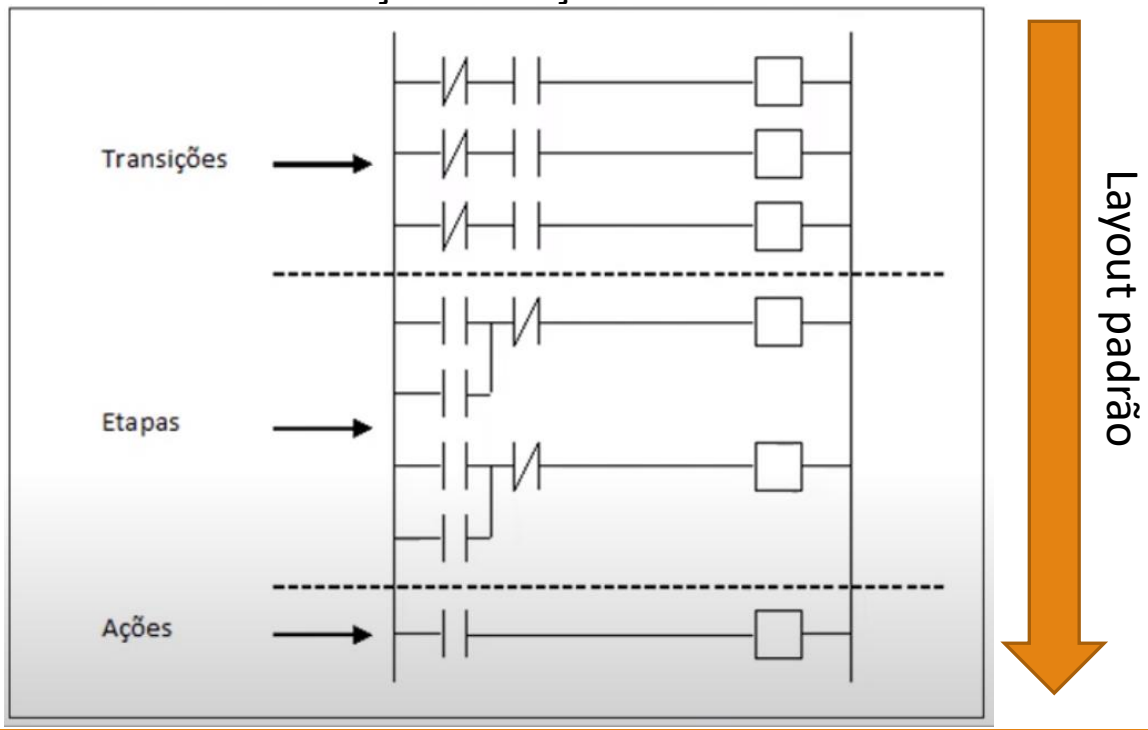


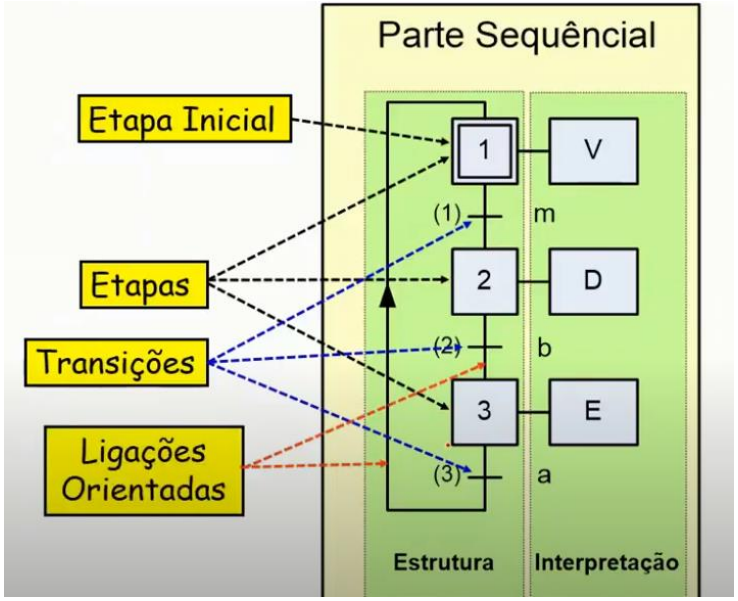
Exemplo de sw para design/simulação: [grafcet studio](), com deploy (upload) do grafcet **diretamente** para CLPs compatíveis (s7-300/400, 1200/1500, codesys v3, raspberry, Arduino Due); Existem outros projetos, inclusive OPEN, com CLPs em Raspberry, Arduino, ESP8266: [Conheça o OpenPLC - O primeiro CLP de Código Aberto Padronizado (embarcados.com.br)]()

Conversão GRAFCET – LADDER - Metodologia

- Para CLP baseado em FPGA, não temos conhecimento de ferramentas de geração automática de conversão grafcet – blocos lógicos, ou grafcet-ladder com blocos lógicos, sendo uma oportunidade de desenvolvimento!

1) Na estrutura do grafcet temos três elementos básicos: **transições, etapas** e as **ações**
2) No Ladder também teremos três partes distintas, nesta ORDEM: transições, depois sequenciamento das etapas e por último a execução das ações





Conversão GRAFCET – LADDER - Metodologia

Exemplo 1: egrafcet.utad.pt/Home/Exemplos_Ex1

Considere um carro que se pode movimentar entre as posições A e B.

A presença do carro em A é representada pela variável Booleana a=1 e em B é representada por b=1. O carro é comandado por um botão m (m=1 quando pressionado).

Inicialmente o carro está na posição A (repouso). As saídas do sistema de controlo são D e E (D=1, o carro desloca-se para a direita; E=1, o carro desloca-se para a esquerda).

Com a utilização de Grafcets vão ser implementados os controladores lógicos que correspondem às seguintes situações:

Sensibilidades/ Entradas	Variável
Botão de acionamento	M (I0)
Fim de curso A	A (I1)
Fim de curso B	B (I2)

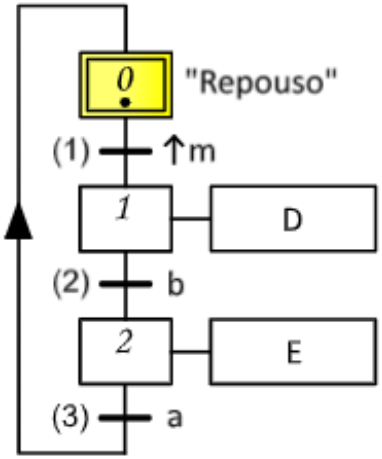
Saídas	Variável
Carro vai para a direta	D (Q0)
Carro vai para a esquerda	E (Q1)

Memorização (variáveis internas) das etapas e transições

Transição	Variável
E0 – E1 (1)	M10
E1 - E2 (2)	M11
E2 - E0 (3)	M12

Etapa	Variável
0	M0
1	M1
2	M2

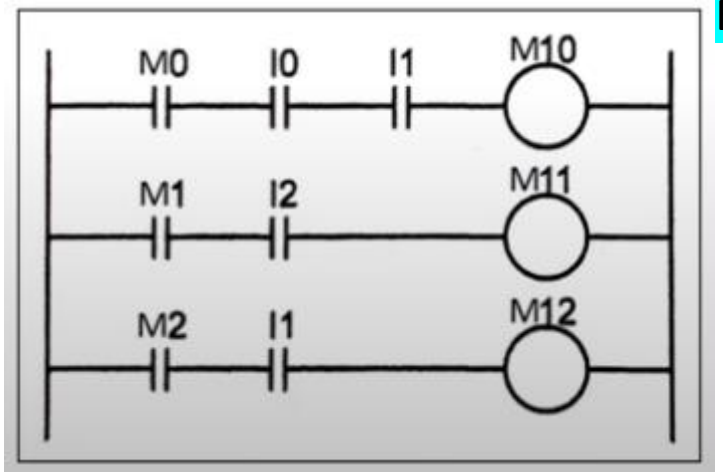
Modelo Grafcet



Conversão GRAFCET – LADDER - Metodologia

Exemplo 1: egrafcet.utad.pt/Home/Exemplos_Ex1

Transições: etapa acima ATIVA e receptividade VERDADEIRA



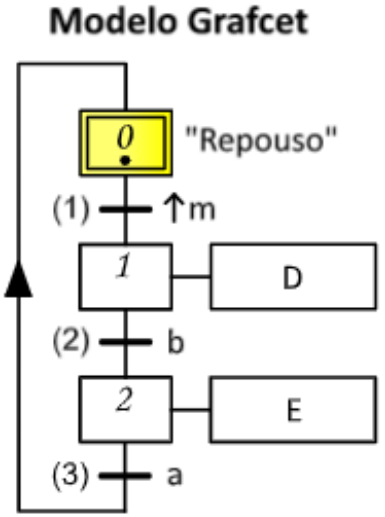
Etapa acima da transição

M10 = M0 AND (I0 AND I1)

Receptividade (condição)

M11 = M1 AND I2

M12 = M2 AND I1

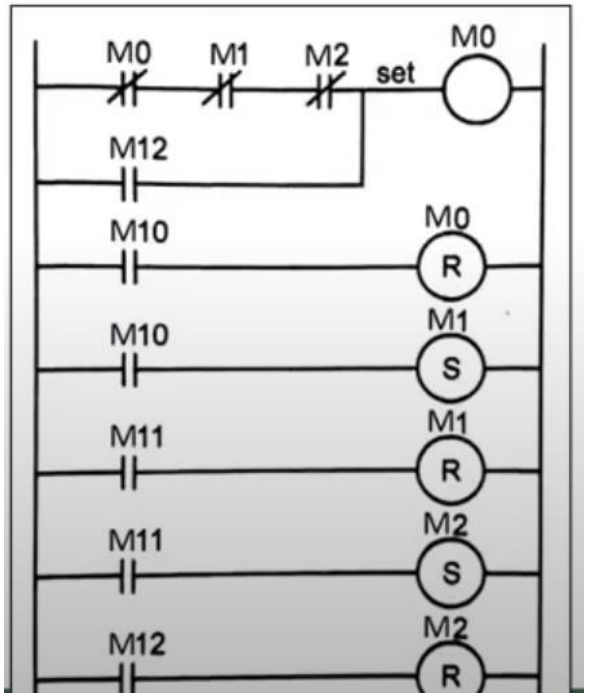


Conversão GRAFCET – LADDER - Metodologia

Exemplo 1: egrafcet.utad.pt/Home/Exemplos_Ex1

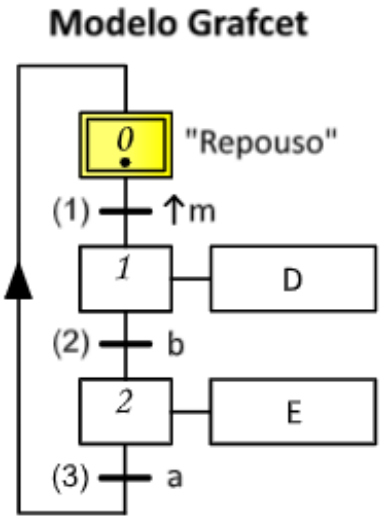
Etapas (estados): a etapa pode necessitar ficar ativa por mais tempo que a transição que a ativa, normalmente pulsante. Logo existem soluções com SET/RESET **e com SELO**

LÓGICA SET/RESET



Qual transição ativa etapa 0 (M0)? = M12 (2->0)
Outra condição INICIAL é não haver etapas ativas, para então SETAR M0 como INICIAL, logo, faz-se a conjunção **NOT(M0) AND NOT(M1) AND NOT(M2)**

Transição que faz chegar em Etapa e Transição **que faz sair da Etapa**

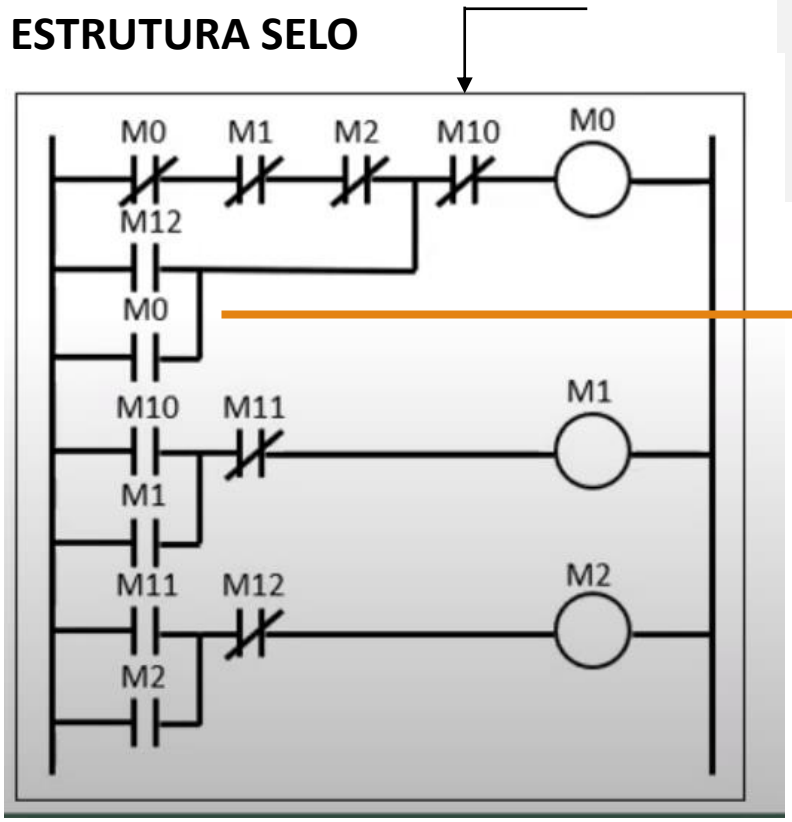


Conversão GRAFCET – LADDER - Metodologia

Exemplo 1: egrafcet.utad.pt/Home/Exemplos_Ex1

Etapas (estados): a etapa pode necessitar ficar ativa por mais tempo que a transição que a ativa, normalmente pulsante. Logo existem soluções com SET/RESET **e com SELO**

ESTRUTURA SELO



Este AND com a transição 0->1 NF, indica que se a transição ocorrer (M10=1, logo NOT(M10)=0), M0 = 0, como se fosse um botão de STOP. Este padrão repete-se para as outras etapas, sempre com a transição de saída da etapa em NF

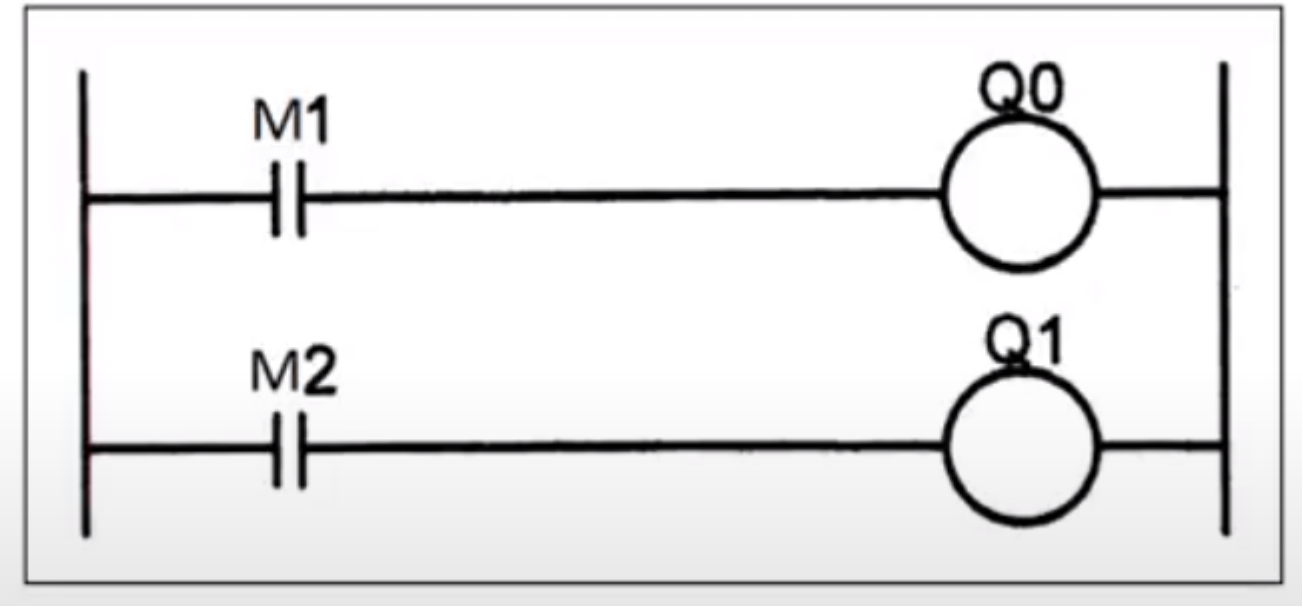
Selo mantém memória (etapa) ativa = 1

Esquema geral: memoriaEtapa =
= (transicaoChegada OR memoriaEtapa) AND NOT (transicaoSaida)

Conversão GRAFCET – LADDER - Metodologia

Exemplo 1: egrafcet.utad.pt/Home/Exemplos_Ex1

Saídas: as saídas equivalentes às respectivas etapas, quando ativas, são disparadas

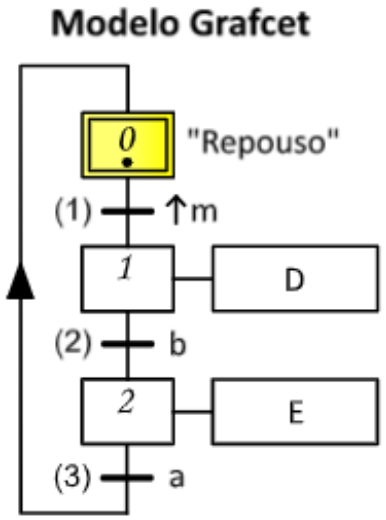


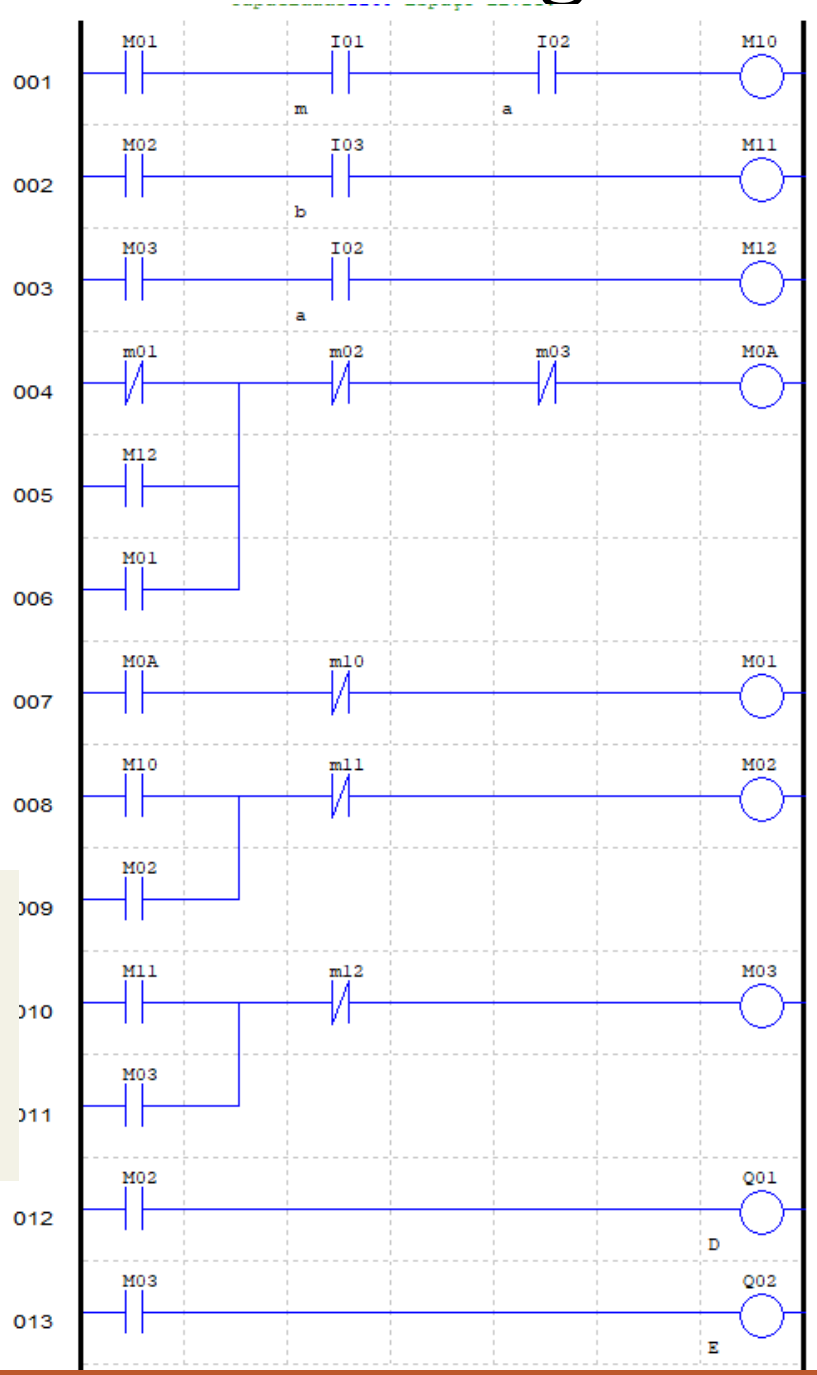
Exercício 1: montagem deste diagrama Ladder no Clic02. Ver exemplo2.cli em embeddedsystems/ladderdiagrams at master · josenalde/embeddedsystems (github.com)

Conversão GRAFCET – LADDER - Metodologia

Exemplo 1: egrafcet.utad.pt/Home/Exemplos_Ex1

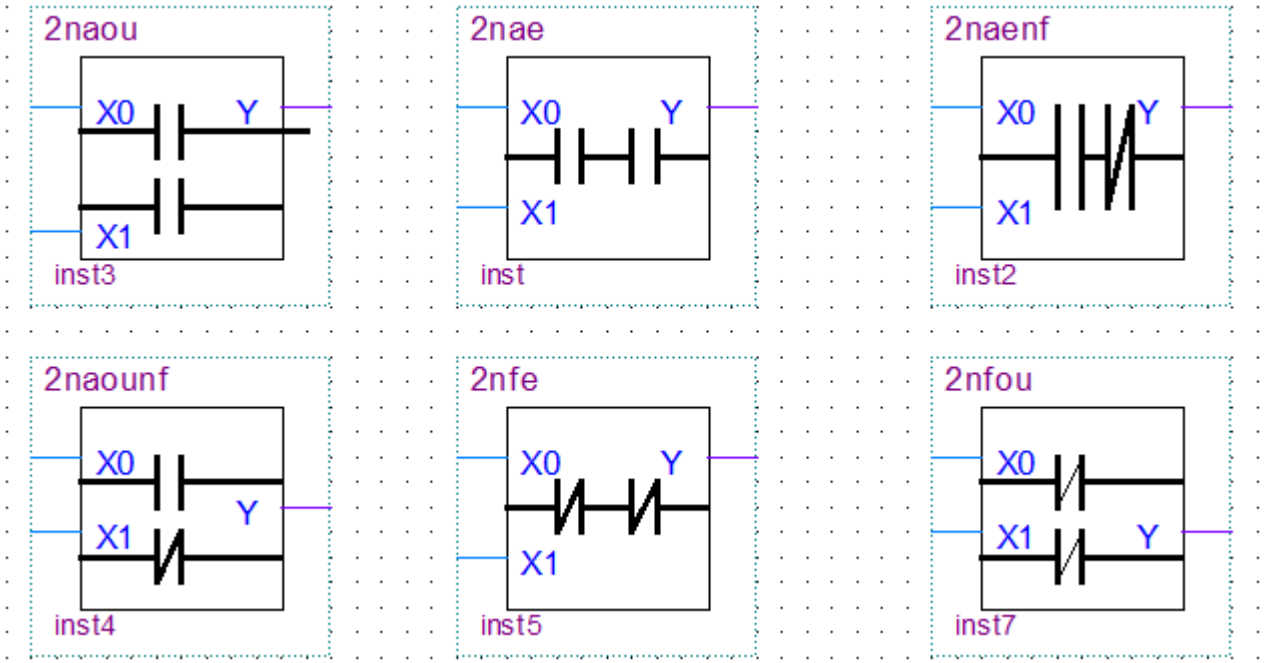
Exercício 1: montagem deste diagrama Ladder no Clic02. Ver exemplo2.cli em embeddedsystems/ladderdiagrams at master · josenalde/embeddedsystems (github.com)





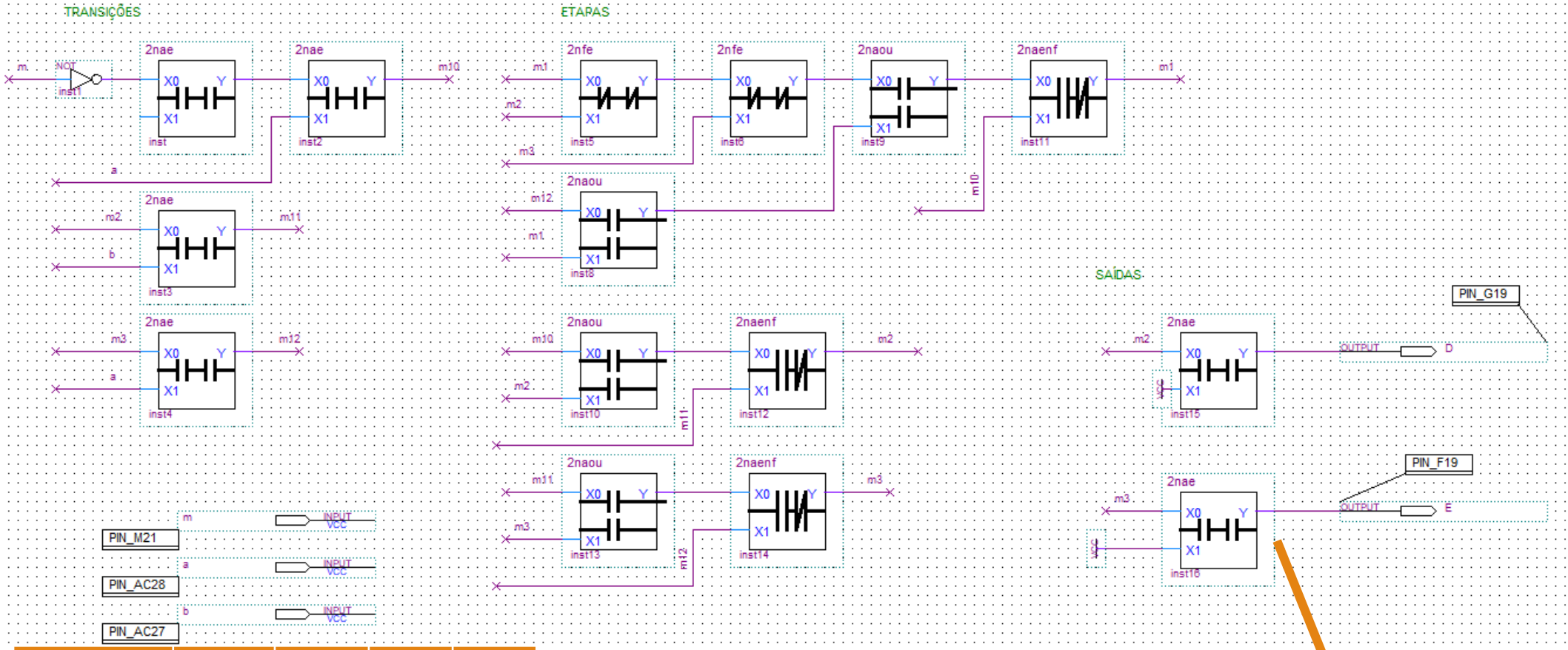
Para escrever esta solução em FPGA, precisamos de blocos lógicos (macrofunções) que representem os relés NA, NF e respectivas combinações AND e OR. Nossa referência propõe os seguintes blocos, mas podemos criar tantos quanto precisarmos

Bloco (.bsf – block symbol files, com respectivos .bdf (block diagram files) Quartus



Para escrever esta solução em FPGA, precisamos de blocos lógicos (macrofunções) que representem os relés NA, NF e respectivas combinações AND e OR. Nossa referência propõe os seguintes blocos, mas podemos criar tantos quanto Precisarmos. Uma opção é colocar tais arquivos .bdf e .bsf numa pasta com nome sugestivo, como lib ou include, e referenciar na criação do projeto

Projeto carro1



Sensibilidades/ Entradas	Variável	Chave PLACA	PINO DE2-115	PINO DE2
Botão de acionamento	M (I0)	KEY[1]	M21	N23
Fim de curso A	A (I1)	SW[1]	AC28	N26
Fim de curso B	B (I2)	SW[2]	AC27	P25

Saídas	Variável	LED	PINO DE2-115	PINO DE2
Carro vai para a direta	D (Q0)	LEDR[0]	G19	AE23
Carro vai para a esquerda	E (Q1)	LEDR[1]	F19	AF23

Y = X0.1 = X0 (NA)
E = m3.1 = m3