

Memória do PLC

Área de memória do CPU

Distribuição das áreas de memória:

A memória do CPU pode ser dividida em 3 áreas: *Load Memory*, *Work Memory* e *System Memory*.

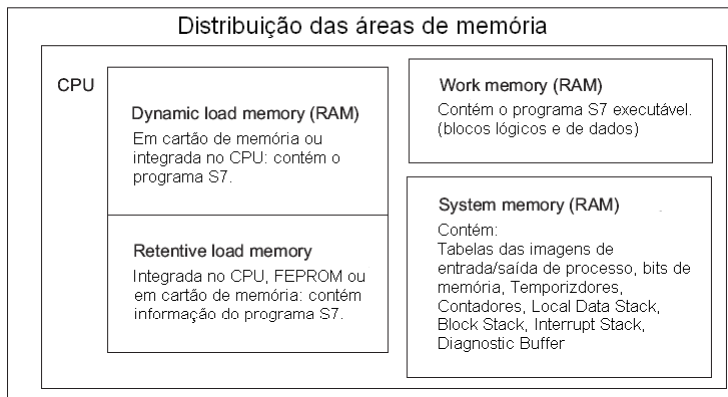
- **Load Memory** (memória de carregamento) guarda todo o programa de utilizador à excepção dos simbólicos e comentários que ficam retidos na consola de programação. Esta memória pode ser volátil (RAM) ou não volátil (EPROM).

- **Work Memory** (memória de trabalho) contém os blocos de programa indispensáveis para executar o programa. O programa é executado apenas nas áreas de memória de trabalho e de sistema.

- **System Memory** (memória de sistema) contém todos os elementos disponibilizados pelo CPU, a saber: tabelas das imagens de entradas de processo e saídas de processo, bits de memória, contadores e temporizadores, *block stack* e *interrupt stack*. Além disso, é também nesta área que está a memória temporária (*local data stack*). A *local data stack* contém a informação temporária dos blocos de programa e apenas está atribuída a cada bloco enquanto este está aberto.

Ao fazer Download para o CPU apenas os blocos lógicos e os blocos de dados são enviados para o CPU. As tabelas de símbolos, comentários e tabelas de variáveis ficam na consola de programação.

Distribuição das áreas de memória do CPU



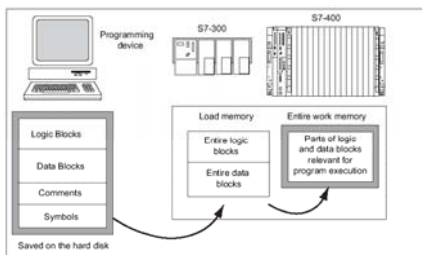
 BACK FORWARD 

Programa de utilizador

Selecionando Programa de utilizador

Para melhorar o desempenho na execução do programa e para evitar uma sobrecarga desnecessária na *work memory* (que não pode ser expandida) só as partes relevantes do programa são carregadas na *work memory*. As partes dos blocos que não são essenciais à execução do programa ficam retidas na *load memory* (por ex. cabeçalhos dos blocos).

A figura mostra o *download* do programa para o CPU.



Os blocos de dados identificados pelo S7 como **UNLINKED** ficam catalogados como não relevantes para a execução do programa. Isto significa que ao fazer o *download* para o CPU, ficam guardados na *Load Memory*.

No entanto é possível a transferência destes blocos de dados para a memória de trabalho utilizando a SFC20 BLKMOV disponível no catálogo:

Libraries → Standard library → System Function Blocks

 BACK FORWARD 

Load memory (memória de carregamento)

Estrutura

A Load Memory integrada do PLC S7-300 é uma parte do tipo RAM e outra do tipo EPROM e pode ser expandida utilizando, para isso, um cartão de memória. A área retentiva da *Load Memory* deve ser configurada nas propriedades do CPU.

No PLC S7-400 é necessário utilizar um cartão de memória para expandir a Load Memory porque a memória integrada é uma RAM usada pelo CPU para gestão interna dos blocos. Alguns CPU's S7-400 mais recentes admitem memória de trabalho adicional.

Os programas guardados na RAM são eliminados quando o programador executa um *Memory Reset* (MRES) ao CPU ou quando é retirado o cartão de memória do CPU.
 Os programas guardados nos cartões de memória EPROM não são apagados com o MRES e permanecem no cartão mesmo sem qualquer alimentação ao PLC.

◀ BACK FORWARD ▶

System Memory (memória de sistema)

Áreas da Memória de Sistema

A memória de sistema está dividida em sub-áreas endereçáveis que podem ser acedidas directamente, utilizando para isso as instruções de programação disponíveis. As tabelas seguintes mostram as sub-áreas disponíveis e a forma de acesso:

Área endereço	Tamanho do endereço	Notação S7 (IEC)	Descrição
Process image input table	Input (bit)	I	No início do ciclo de programa, é feita a leitura dos módulos de entrada e é actualizada a tabela de Imagem de Entradas de Processo
	Input byte	IB	
	Input word	IW	
	Input double word	ID	
Process image output table	Output (bit)	Q	Durante a execução de um ciclo de programa, o CPU vai actualizando esta tabela (Imagem de Saídas de Processo). No final do ciclo, o CPU envia a informação desta tabela para os módulos de saídas.
	Output byte	QB	
	Output word	QW	
	Output double word	QD	

◀ BACK FORWARD ▶

System Memory (memória de sistema)

Área endereço	Tamanho do endereço	Notação S7 (IEC)	Descrição
Bit memory	Memory (bit)	M	Esta é a área de armazenamento para as memórias utilizadas no programa de utilizador.
	Memory byte	MB	
	Memory word	MW	
	Memory double word	MD	
Timers	Timer (T)	T	Área reservada para Temporizadores
Counters	Counter (C)	C	Área de memória reservada para contadores
Data block	Data block, opened with "OPN DB":	DB	Os Data Blocks contêm informação utilizada no programa. Há dois tipos de Data Blocks: Os primeiros podem ser definidos para uso geral e podem ser acedidos por todos os blocos lógicos. São chamados blocos de dados de uso geral (DB). Os segundos, estão associados a uma FB ou SFB específica e apenas podem ser acedidos por esta. São chamados blocos de dados de instância (DBI).
	Data bit	DBX	
	Data byte	DBB	
	Data word	DBW	
	Data double word	DBD	
	Data block, opened with "OPN DI":	DI	
	Data bit	DIX	
	Data byte	DIB	
	Data word	DIW	
	Data double word	DID	

◀ BACK FORWARD ▶

System Memory (memória de sistema)

Área endereço	Tamanho do endereço	Notação S7 (IEC)	Descrição
Local data	Local data bit	L	Esta área contém a informação temporária do bloco que está a ser executado.
	Local data byte	LB	
	Local data word	LW	
	Local data double word	LD	
Peripheral (I/O) area: inputs	Peripheral input byte	PIB	Esta área permite o acesso directo aos módulos de entrada.
	Peripheral input word	PIW	
	Peripheral input double word	PID	
Peripheral (I/O) area: outputs	Peripheral output byte	PQB	Esta área permite o acesso directo aos módulos de saída.
	Peripheral output word	PQW	
	Peripheral output double word	PQD	

◀ BACK FORWARD ▶

System Memory (memória de sistema)

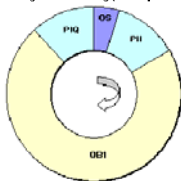
Tabelas de Imagens de entradas/saídas de processo

Ao utilizar no programa as áreas de endereço de entrada (I) ou de saída (Q) o sistema operativo não acede directamente aos módulos de entradas ou saídas digitais. Vai, em vez disso, recolher os valores numa área da *System Memory* chamada **Process Image**.

Esta área é actualizada em todos os ciclos do programa:

No início do ciclo, o sistema operativo lê o estado eléctrico dos módulos de entradas digitais e escreve os valores na tabela de imagem de entradas de processo (PII). Ao concluir esta tarefa, são executados todos os blocos do programa e é actualizada a tabela de imagem de saídas de processo (PIQ). No final da execução do programa, o sistema operativo procede à leitura da tabela de imagem de saídas de processo e envia os sinais eléctricos para os módulos de saídas digitais. A principal vantagem desta utilização é a consistência das entradas/saídas durante o tempo de um ciclo de programa. Este esquema de funcionamento está resumido na figura:

Cyclic Program Processing (CPU's up to 10/88)



◀ BACK FORWARD ▶

System Memory (memória de sistema)

Local Data Stack (L Stack)

A Lstack providencia espaço de memória para:

- Variáveis temporárias dos blocos de programa
- Informação inicial dos blocos de organização (OB)
- Informação sobre transferência de parâmetros
- Resultados intermédios das instruções dos programas em LADDER

Ao utilizar blocos de organização (OB) é possível declarar e utilizar variáveis temporárias (TEMP). O espaço de memória destas variáveis apenas está disponível enquanto o bloco está aberto. Este espaço é depois reutilizado para o próximo bloco a ser aberto. Todos os blocos de organização necessitam de 20 bytes de memória (L Stack) para guardar a sua informação própria de inicialização.

O espaço de memória para a L Stack varia de acordo com o CPU utilizado. Este espaço de memória é limitado. No entanto, é também dividido pelas classes de prioridade, ou seja, se um bloco de organização chama outro e assim sucessivamente, há sempre uma área de memória igual (256 bytes) reservada para cada um dos blocos, pois nenhum deles foi concluído.

Para os CPU's S7-318 e S7-4XX é possível ajustar este espaço de memória para cada classe de prioridade.

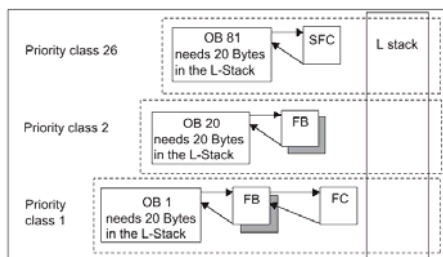
◀ BACK FORWARD ▶

System Memory (memória de sistema)

Local Data Stack (L Stack)

A figura exemplifica a situação em que o OB1 é interrompido pelo OB20, que por sua vez é interrompido pelo OB81. As classes de prioridade são diferentes para cada um destes OB's. É necessário ter em atenção que:

- Todas as variáveis temporárias dos blocos de organização e blocos lógicos associados estão guardadas na L Stack. Por isso, se o encadeamento de blocos for muito extenso, a Stack Local poderá entrar em *Overflow*, o que provoca a paragem da execução do programa (O CPU entra em *Stop Mode*)



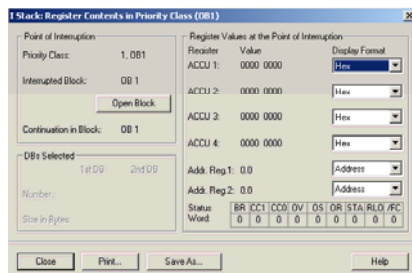
BACK FORWARD

System Memory (memória de sistema)

Interrupt Stack (I Stack)

Quando a execução do programa é interrompida por um bloco (OB) prioritário, é necessário guardar alguma informação. Assim, o Sistema Operativo guarda na I Stack a informação existente nos Acumuladores e Registos de Endereço. Guarda também informação sobre o Bloco de Programa. No final da interrupção, é reposta a informação e é assim possível prosseguir com o programa a partir do ponto onde foi interrompido.

Quando o CPU entra em STOP, é possível visualizar a informação da I Stack com a ferramenta do editor LAD/STL/FBD: **PLC → Module Information → Stacks**



Quando o CPU entra em Stop, o conteúdo da I Stack pode ajudar o programador a identificar a causa da paragem do CPU

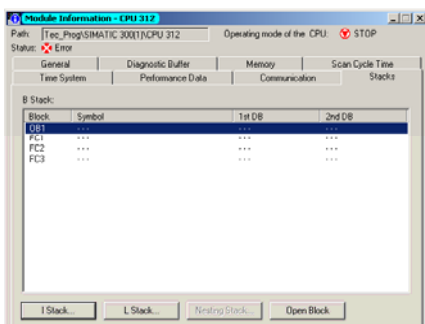
BACK FORWARD

System Memory (memória de sistema)

Block Stack (B Stack)

Quando o processamento de um bloco é interrompido, seja pela chamada de outro bloco ou devido a um bloco de interrupção prioritário, a B Stack guarda informação sobre o número e tipo de bloco interrompido e se existir, o bloco de dados (DB ou DI) aberto no momento da interrupção.

Quando o CPU entra em STOP, é possível visualizar a informação da B Stack com a ferramenta do editor LAD/STL/FBD: **PLC → Module Information → Stacks**



A B Stack lista todos os blocos que não foram totalmente executados no momento da interrupção do CPU

System Memory (memória de sistema)

Diagnostic Buffer

O Buffer de Diagnostico armazena mensagens de sistema pela sua ordem de ocorrência. As mensagens são apresentadas em forma de tabela em que a primeira entrada corresponde ao evento mais recente. A área de memória para esta tabela depende da versão do CPU (por exemplo para o CPU 314 a área armazena 100 eventos). Quando o número máximo de entradas é atingido, a entrada seguinte é colocada na primeira posição da tabela, deslocando todos os outros uma posição abaixo. Assim, a ultima posição da tabela é eliminada.

As mensagens apresentadas destinam-se a dotar o programador de uma ferramenta que lhe possibilite conhecer os eventos de erro ou falha do CPU, a saber:

- Falhas em módulos e ligações
- Erros de sistema e do programa
- Transições de modo de funcionamento do CPU
- Eventos de diagnostico programados pelo utilizador (utilizando a SFC52)

A informação do Buffer de Diagnostico não é apagada pelo utilizador nem mesmo com um *Reset* à memória do CPU.

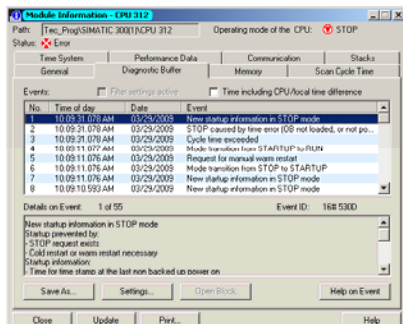
System Memory (memória de sistema)

Diagnostic Buffer

A figura mostra o Buffer de Diagnóstico de um CPU. De notar que é fornecida ao utilizador informação detalhada para cada um dos eventos da tabela. Além disso, o botão *Help on Event* disponibiliza uma ajuda, por vezes muito útil, para o evento em causa.

Esta ferramenta pode ser acedida no editor LAD/STL/FBD em:

PLC → Module Information → Diagnostic Buffer



Áreas de memória retentivas

CPU S7-300

Em caso de falha de alimentação ou ao efectuar um RESET à memória do CPU, toda a informação existente na *Load Memory*, *Work Memory* e *System Memory* é perdida. No entanto, há algumas formas de proteger esta informação em caso de falha de alimentação:

- Bateria de *Backup* (mantém a *Load Memory*, *Work Memory* e parte da *System Memory*)
- Guardar o programa na EPROM (integrada no CPU ou em *Memory Card*)
- Guardar informação na área não volátil **NVRAM** (o tamanho depende da versão do CPU).

A NVRAM é uma área de memória não volátil (mesmo sem baterias de *backup*), que pode ser configurada pelo programador para reter a informação de alguns DB's, Temporizadores, Contadores e Bits de Memória, em caso de falha de alimentação ao CPU. Esta configuração é feita em:

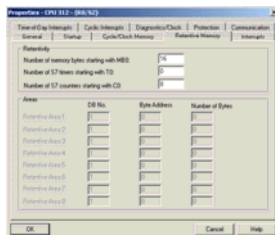
HW Config → CPU Object Properties → Retentive Memory

Para garantir o funcionamento das comunicações depois de uma falha de alimentação, o endereço MPI do CPU também está guardado na área não volátil (NVRAM).

Áreas de memória retentivas

CPU S7-300

A figura representa a caixa de configuração da memória retentiva (NVRAM) do CPU.



CPU S7-400

Em caso de falha de alimentação ao CPU e sem baterias de *Backup*, apenas é memorizado o endereço MPI. Com as baterias de *Backup*, todo o conteúdo da memória RAM do CPU é preservado

Tal como no S7-300, temporizadores, contadores e bits de memória também pode ser configurados como retentivos pelo programador.

Pode consultar informação mais detalhada no manual: "S7-400, M7-400 Programmable Controllers, Module Specifications"

◀ BACK FORWARD ▶

Memória do PLC

Mapa de configuração de memória do PLC




		MB0								MB1							
MD 0	MW0	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
		0.7	0.6	0.5	0.4	0.3	0.2	0.1	0.0	1.7	1.6	1.5	1.4	1.3	1.2	1.1	1.0
		MB2								MB3							
MD 2	MW2	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
		2.7	2.6	2.5	2.4	2.3	2.2	2.1	2.0	3.7	3.6	3.5	3.4	3.3	3.2	3.1	3.0
		MB4								MB5							
MD 4	MW4	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
		4.7	4.6	4.5	4.4	4.3	4.2	4.1	4.0	5.7	5.6	5.5	5.4	5.3	5.2	5.1	5.0
		MB6								MB7							
MD 6	MW6	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
		6.7	6.6	6.5	6.4	6.3	6.2	6.1	6.0	7.7	7.6	7.5	7.4	7.3	7.2	7.1	7.0

◀ BACK FORWARD ▶

Memória do PLC

Var : [VAT_2 -- @FormaçãoIS7 Program(1) ONLINE]

Table Edit Insert PLC Variable View Options Window Help

	Address	Symbol	Display format	Status value	Modify value
1	M/V 0	BIN		2#0000_0001_0000_0000	
2	M/V 2	BIN		2#0000_0000_0000_0001	
3	M/V 4	BIN		2#0010_0000_0000_0000	
4	M/V 6	BIN		2#0000_0000_0000_0000	
5	M/V 8	BIN		2#0000_0000_0000_0000	
6	M/V 10	BIN		2#0000_1111_0000_0000	
7	M/V 12	BIN		2#0000_0000_0000_0000	
8	M/V 14	BIN		2#0000_0000_0000_0000	
9	M/V 16	BIN		2#0000_0000_0000_0000	
10	M/V 18	BIN		2#0000_0000_0000_0000	
11	M/V 20	BIN		2#0000_0000_0000_0000	
12	M/V 22	BIN		2#0000_0000_0100_0000	
13	M/V 24	BIN		2#0000_0000_0000_0000	
14	M/V 26	BIN		2#0000_0000_0000_0000	
15	M/V 28	BIN		2#0000_0000_0000_0000	
16	M/V 30	BIN		2#0000_0000_0000_0000	
17					
18	M 0.0	BOOL		true	true
19	M 3.0	BOOL		true	true
20	MB 10	BIN		2#0000_1111	2#0000_1111
21	M 23.6	BOOL		true	true
22					
23					
24					
25					

⏪ BACK FORWARD ⏩