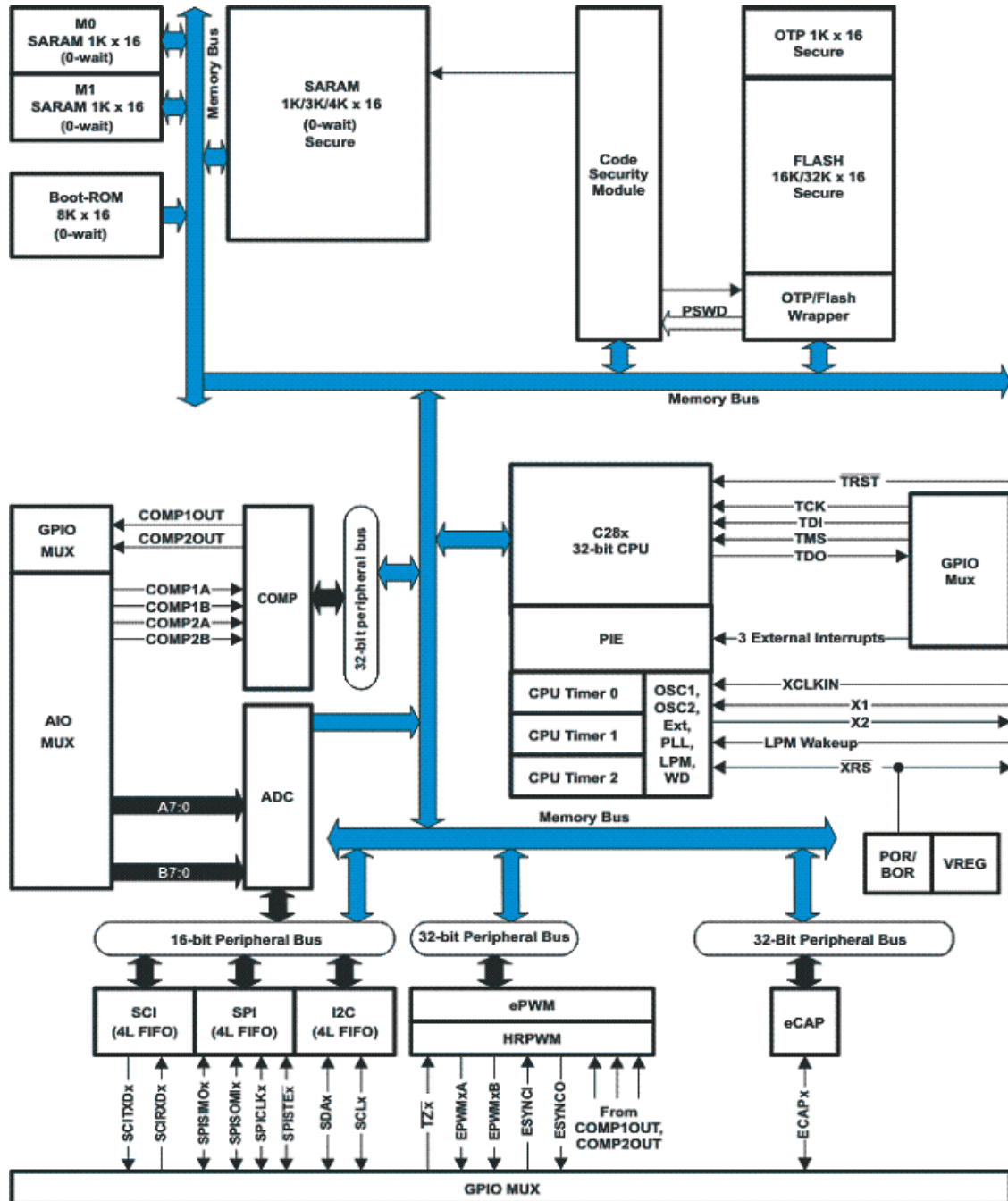


Diagrama de blocos funcional do TMS320F28027



Mapeamento de Memória do TMS320F28027

	Data Space	Prog Space	
Low 64K (24x/240x Equivalent Data Space)	0x00 0000	M0 Vector RAM (Enabled if VMAP = 0)	
	0x00 0040		
	0x00 0400	M0 SARAM (1K x 16, 0-Wait)	
	0x00 0800	M1 SARAM (1K x 16, 0-Wait)	
	0x00 0D00	Peripheral Frame 0	Reserved
		PIE Vector - RAM (256 x 16) (Enabled if VMAP = 1, ENPIE = 1)	
	0x00 0E00	Peripheral Frame 0	
	0x00 2000	Reserved	
	0x00 6000	Peripheral Frame 1 (4K x 16, Protected)	Reserved
	0x00 7000	Peripheral Frame 2 (4K x 16, Protected)	
	0x00 8000	L0 SARAM (4K x 16) (0-Wait, Secure Zone + ECSL, Dual Mapped)	
	0x00 9000	Reserved	
	0x3D 7800	User OTP (1K x 16, Secure Zone + ECSL)	
	0x3D 7C00	Reserved	
0x3D 7C80	Calibration Data		
0x3D 7CC0	Reserved		
0x3D 8000	Reserved		
High 64K (24x/240x Equivalent Program Space)	0x3F 0000	FLASH (32K x 16, 4 Sectors, Secure Zone + ECSL)	
	0x3F 7FF8	128-Bit Password	
	0x3F 8000	L0 SARAM (4K x 16) (0-Wait, Secure Zone + ECSL, Dual Mapped)	
	0x3F 9000	Reserved	
	0x3F E000	Boot ROM (8K x 16, 0-Wait)	
	0x3F FFC0	Vector (32 Vectors, Enabled if VMAP = 1)	

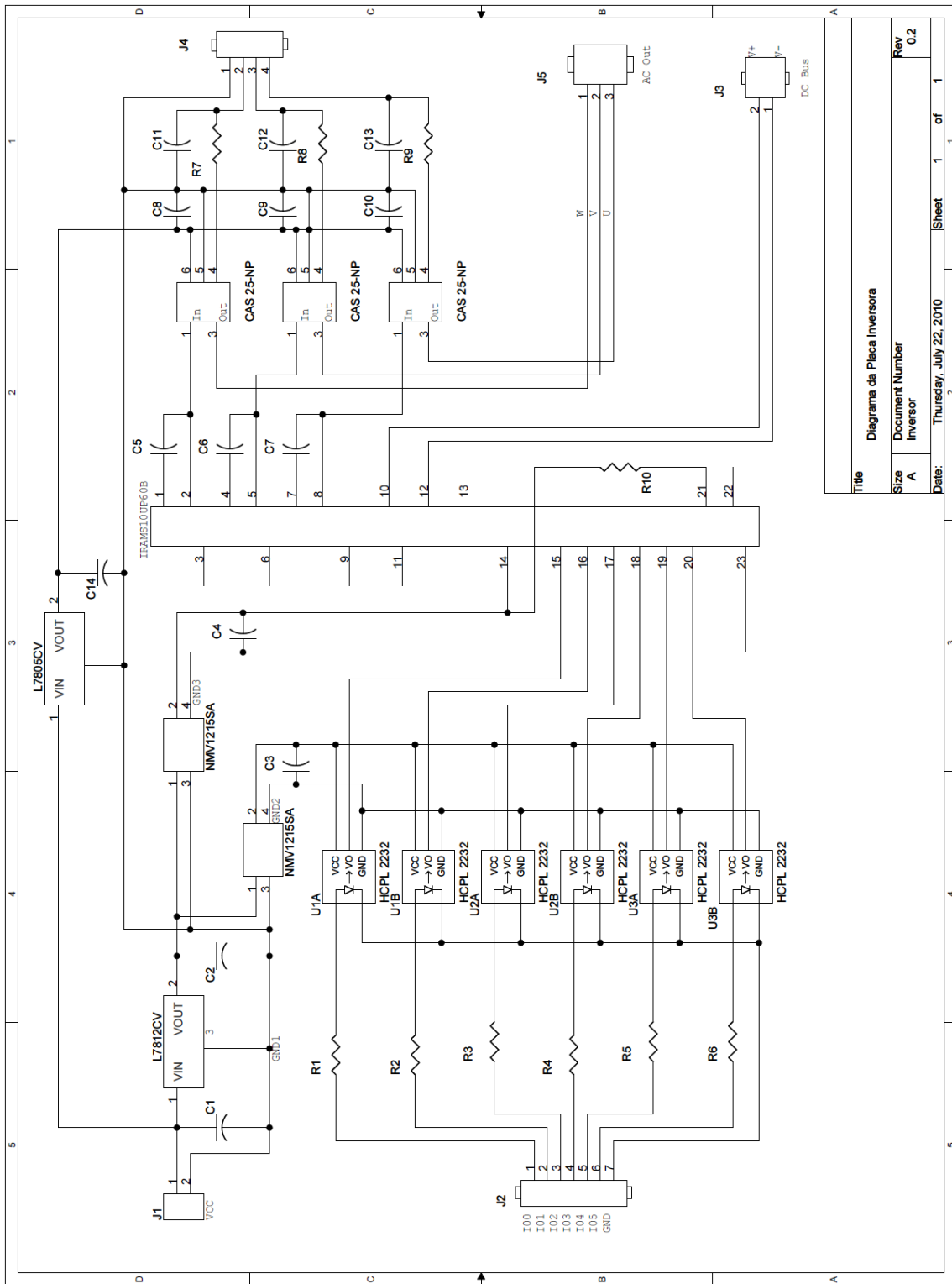
Exemplo de programação do módulo ePWM do TMS320F28027

```

//=====
// Configuration
//=====
// Initialization Time
//=====// EPWM Module 1 config
EPwm1Regs.TBPRD = 800; // Period = 1600 TBCLK counts
EPwm1Regs.TBPHS.half.TBPHS = 0; // Set Phase register to zero
EPwm1Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN; // Symmetrical mode
EPwm1Regs.TBCTL.bit.PHSEN = TB_DISABLE; // Master module
EPwm1Regs.TBCTL.bit.PRDL = TB_SHADOW;
EPwm1Regs.TBCTL.bit.SYNCSEL = TB_CTR_ZERO; // Sync down-stream module
EPwm1Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on CTR=Zero
EPwm1Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // load on CTR=Zero
EPwm1Regs.AQCTLA.bit.CAU = AQ_SET; // set actions for EPWM1A
EPwm1Regs.AQCTLA.bit.CAD = AQ_CLEAR;
EPwm1Regs.DBCTL.bit.MODE = DB_FULL_ENABLE; // enable Dead-band module
EPwm1Regs.DBCTL.bit.POLSEL = DB_ACTV_HIC; // Active Hi complementary
EPwm1Regs.DBFED = 50; // FED = 50 TBCLKs
EPwm1Regs.DBRED = 50; // RED = 50 TBCLKs
// EPWM Module 2 config
EPwm2Regs.TBPRD = 800; // Period = 1600 TBCLK counts
EPwm2Regs.TBPHS.half.TBPHS = 0; // Set Phase register to zero
EPwm2Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN; // Symmetrical mode
EPwm2Regs.TBCTL.bit.PHSEN = TB_ENABLE; // Slave module
EPwm2Regs.TBCTL.bit.PRDL = TB_SHADOW;
EPwm2Regs.TBCTL.bit.SYNCSEL = TB_SYNC_IN; // sync flow-through
EPwm2Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm2Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm2Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on CTR=Zero
EPwm2Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // load on CTR=Zero
EPwm2Regs.AQCTLA.bit.CAU = AQ_SET; // set actions for EPWM2A
EPwm2Regs.AQCTLA.bit.CAD = AQ_CLEAR;
EPwm2Regs.DBCTL.bit.MODE = DB_FULL_ENABLE; // enable Dead-band module
EPwm2Regs.DBCTL.bit.POLSEL = DB_ACTV_HIC; // Active Hi complementary
EPwm2Regs.DBFED = 50; // FED = 50 TBCLKs
EPwm2Regs.DBRED = 50; // RED = 50 TBCLKs
// EPWM Module 3 config
EPwm3Regs.TBPRD = 800; // Period = 1600 TBCLK counts
EPwm3Regs.TBPHS.half.TBPHS = 0; // Set Phase register to zero
EPwm3Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN; // Symmetrical mode
EPwm3Regs.TBCTL.bit.PHSEN = TB_ENABLE; // Slave module
EPwm3Regs.TBCTL.bit.PRDL = TB_SHADOW;
EPwm3Regs.TBCTL.bit.SYNCSEL = TB_SYNC_IN; // sync flow-through
EPwm3Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm3Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm3Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on CTR=Zero
EPwm3Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // load on CTR=Zero
EPwm3Regs.AQCTLA.bit.CAU = AQ_SET; // set actions for EPWM3A
EPwm3Regs.AQCTLA.bit.CAD = AQ_CLEAR;
EPwm3Regs.DBCTL.bit.MODE = DB_FULL_ENABLE; // enable Dead-band module
EPwm3Regs.DBCTL.bit.POLSEL = DB_ACTV_HIC; // Active Hi complementary
EPwm3Regs.DBFED = 50; // FED = 50 TBCLKs
EPwm3Regs.DBRED = 50; // RED = 50 TBCLKs
// Run Time (Note: Example execution of one run-time instant)
//=====
EPwm1Regs.CMPA.half.CMPA = 500; // adjust duty for output EPWM1A
EPwm2Regs.CMPA.half.CMPA = 600; // adjust duty for output EPWM2A
EPwm3Regs.CMPA.half.CMPA = 700; // adjust duty for output EPWM3A

```


Esquema eléctrico do Módulo de Potência



Title	Diagrama da Placa Inversora		
Size	A	Document Number	Rev 02
Date:	Thursday, July 22, 2010	Sheet	1 of 1