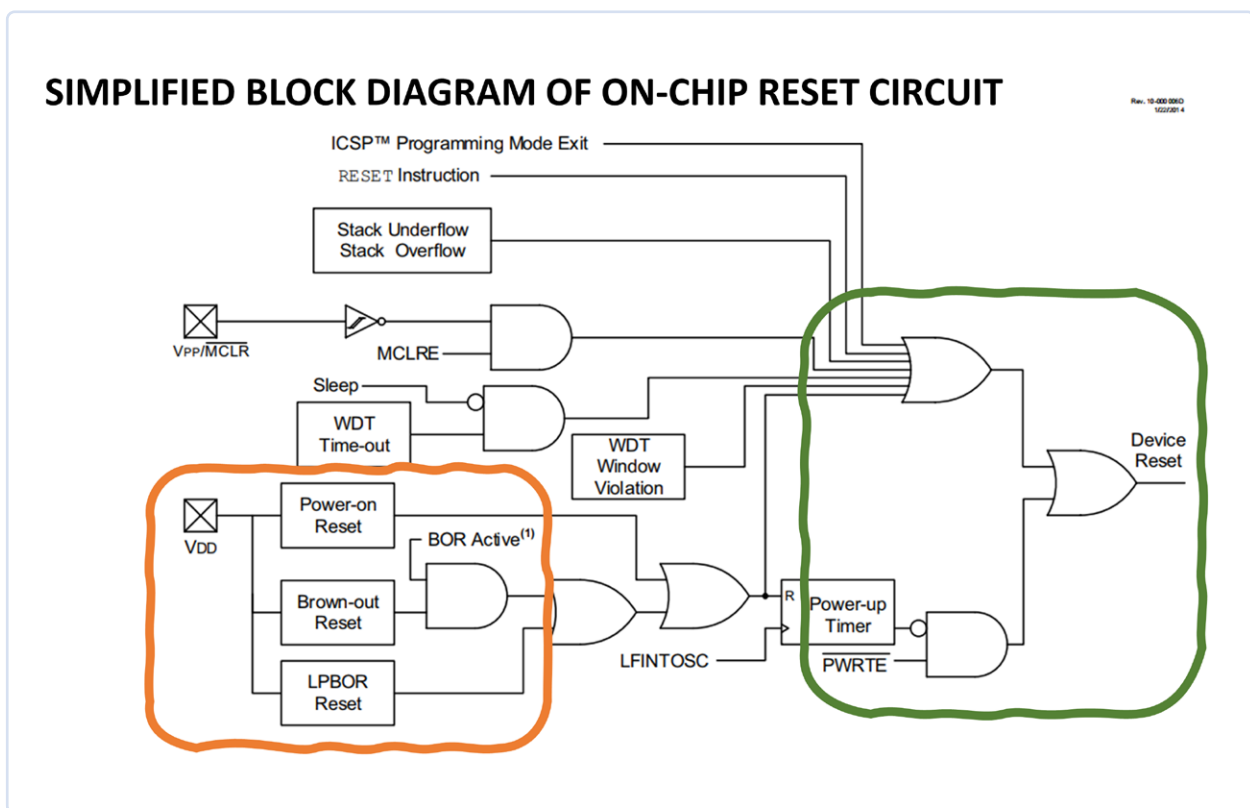


Microcontroller-documentatie verklaard

deel 3: blokschema's en meer



Figuur 1. De resetschakeling biedt zeer weinig configuratiemogelijkheden, en sommige reset-mogelijkheden zijn gekoppeld aan de V_{DD} -voedingspin (bron: Microchip Technology).

Stuart Cording (Elektor)

In de vorige delen van deze serie [1] hebben we de registerfunctionaliteit behandeld en het blokschema van de klok bekeken. We geven hier meer blokschema's en kijken waar we de rest van de benodigde documentatie kunnen vinden.

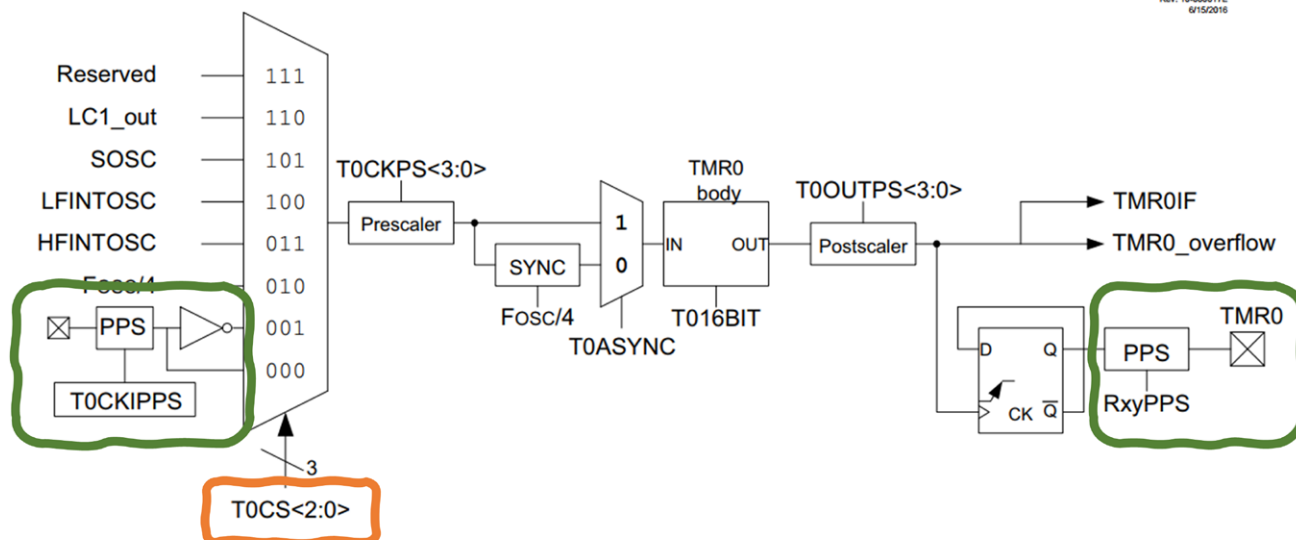
We blijven bij de 8-bit PIC16F18877-microcontroller van Microchip Technology als voorbeeld, en het is raadzaam om de datasheet [2] te downloaden om deze uitleg te volgen.

Herstarten met een reset

Als de klok-periferie het belangrijkste onderdeel van een microcontroller is, dan is de implementatie van de resetschakeling

BLOCK DIAGRAM OF TIMER0

Rev. 10-000017E
6/15/2016



Figuur 2. Het blokschema voor **TIMER0** laat zien dat er een aantal bronnen voor het kloksignaal is, en dat hij zelfs een uitgangssignaal kan genereren naast interne interruptsignalen (bron: Microchip Technology).

het op één na belangrijkste onderdeel. Net als de klok hoeft deze maar één keer geconfigureerd te worden (als er al configuratiemogelijkheden zijn), maar het bespaart later een hoop ongemak als je goed begrijpt waardoor een reset kan worden veroorzaakt. In het voorbeeld op pagina 100 (**figuur 1**) zien we dat twee pinnen (gemarkeerd als vierkantjes met een kruisje, aan de linkerkant) een reset kunnen genereren. Dit als aanvulling op allerlei andere interne manieren. Eén daarvan is de **RESET**-instructie, twee komen van de stack under- en overflow, terwijl andere gekoppeld zijn aan de watchdog-timer.

We zien een aantal reset-mechanismen (oranje gemarkeerd) die zijn gekoppeld aan de voedingsaansluiting van de microcontroller, zoals de power-on en brown-out detectie. Bij deze controller zijn er hier weinig mogelijkheden om in te grijpen, waarbij in principe alle interruptbronnen potentiële resetsignalen zijn (groen gemarkeerd). Zoals we eerder zagen bij het **STATUS**-register, kan de oorzaak van een vorige reset in sommige gevallen worden achterhaald.

Blokschema's van de periferie

De blokschema's voor de periferie zijn even divers als complex. Hier hebben we een deel van een eenvoudige timer (**TIMER0**, op pagina 396) gekozen (**figuur 2**). Aangegeven in oranje zien we opnieuw het gebruik van een groep bits in (vermoedelijk één) register om te kiezen tussen een aantal ingangsbronnen voor deze **TIMER0**-periferie. In groen gemarkeerd zien we ook dat één bron een IC-pin kan zijn, terwijl het blok ook een signaal naar een pin kan sturen. Mogelijk herken je de D flip-flop [3] aan de rechterkant die verbonden is met de uitgangspin, maar veel van de blokken in het schema zijn weinig meer dan vierkanten. Een paar daarvan, zoals de pre- en postscaler, spreken voor zich. Het **SYNC**-blok lijkt vanzelfsprekend, maar een ontwerper zal waarschijnlijk de begeleidende tekst nauwkeurig moeten lezen om de functie ervan helemaal te begrijpen. Schema's en tekst gaan dus vaak hand in hand in datasheets. Soms is het nodig wat testcode te schrijven om echt te begrijpen hoe sommige periferie werkt.

Het bouwen van een board

Op een gegeven moment moet het ontwerp op een print worden gezet als je van plan bent je ontwerp op grotere schaal te produceren. De meeste computerondersteunde ontwerpsoftware (CAD) zal de schemasymbolen voor je processor hebben, evenals de footprint ('land pattern') voor verschillende behuizingen voor het ontwerp van de print. Maar als die ontbreken, bevat de datasheet waarschijnlijk technische tekeningen voor elke behuizing en de bijbehorende footprint. In dit geval zijn ze te vinden op pagina 639.

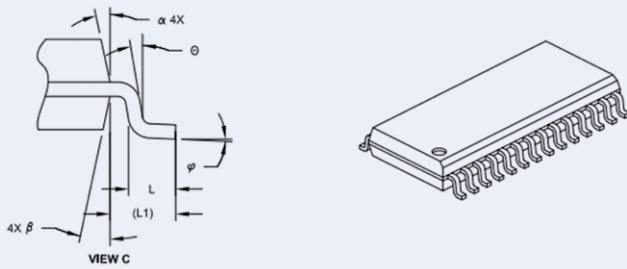
De SOIC-behuizing in **figuur 3** is te vinden op pagina's 645 en 646. Voor de ontwerper van de print is de hoogte van de behuizing tegenwoordig misschien wel het belangrijkste gegeven als het ontwerp zo compact mogelijk moet blijven (de breedte en lengte zijn al bekend). Veel behuizingen hebben tegenwoordig ook een 'heat dissipation pad', dat is een apart vlak dat op de print gesoldeerd moet worden om de warmteafvoer te bevorderen. Dat is hier niet het geval, maar als dat wel het geval was, moeten er aanwijzingen te vinden zijn hoe dit gedaan moet worden. Misschien moet je ook controleren of dit pad met de massa van de chip is verbonden, of wellicht met een andere pin.

Wat staat er maar zelden in de datasheet van een microcontroller?

Zoals je inmiddels weet, staat er veel informatie in een datasheet. Er zijn echter een paar dingen die misschien zijn weggelaten de datasheet anders te dik wordt, of omdat de informatie bij veel onderdelen voorkomt en afzonderlijke documentatie verdient. Bijvoorbeeld:

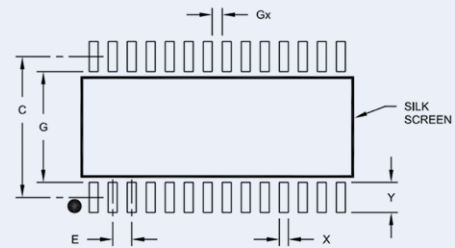
- > de instructieset van de processor: als het aantal instructies beperkt is (minder dan 60...70), staan ze misschien allemaal met uitleg in de datasheet. Zo niet, dan is er waarschijnlijk een afzonderlijk document waarin elke instructie in detail wordt uitgelegd;
- > codevoorbeelden: deze worden vaker afgedrukt als de microcontroller nog gewoon in assembler wordt geprogrammeerd.

28-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]



Dimension	Limits	MILLIMETERS		
		MIN	NOM	MAX
Number of Pins	N	28		
Pitch	e	1.27 BSC		
Overall Height	A	-	-	2.65
Molded Package Thickness	A2	2.05	-	-
Standoff	S	0.10	-	0.30
Overall Width	E	10.30 BSC		
Molded Package Width	E1	7.50 BSC		
Overall Length	D	17.90 BSC		
Chamfer (Optional)	h	0.25	-	0.75
Foot Length	L	0.40	-	1.27
Footprint	L1	1.40 REF		
Lead Angle	theta	0°	-	-
Foot Angle	phi	0°	-	8°
Lead Thickness	c	0.18	-	0.33
Lead Width	b	0.31	-	0.51
Mold Draft Angle Top	alpha	5°	-	15°
Mold Draft Angle Bottom	beta	5°	-	15°

28-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]



Dimension	Limits	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E	1.27 BSC		
Contact Pad Spacing	C	9.40		
Contact Pad Width (X28)	X	0.60		
Contact Pad Length (X28)	Y	2.00		
Distance Between Pads	Gx	0.67	-	-
Distance Between Pads	G	7.40	-	-

Figuur 3. De tekeningen van de behuizing geven alle afmetingen (links) samen met het aanbevolen 'land pattern' voor de print (rechts) (bron: Microchip Technology).

Voorbeelden in hogere talen zoals C hebben weinig zin, omdat de compiler de precieze instructies bepaalt die gebruikt worden;

- schemavoorbeelden: deze komen het meest voor met betrekking tot functionaliteit die uniek is voor de betreffende microcontroller, zoals de oscillator of de voeding, of complexere interfaces die specifieke eisen stellen de belasting van signalen, zoals USB. Het is echter waarschijnlijker dat deze in meer detail worden behandeld in een 'application note' over dat onderwerp.

Wat staat er niet in de datasheet van een microcontroller?

De volgende onderwerpen worden gewoonlijk niet in een datasheet vermeld, maar in andere documentatie. Dit is meestal omdat het een onderwerp betreft dat een groot aantal microcontrollers betreft.

- het flashen van het geheugen van de microcontroller: dit wordt normaal gesproken apart behandeld als onderwerp voor degenen die zich richten op het programmeren van microcontrollers voor massaproductie;
- het gebruik van ontwikkeltools: de compiler, het gebruik van een Integrated Development Environment (IDE) en debug-tools hebben hun eigen documentatie;
- een gedetailleerde uitleg van de processorkern: dit wordt vaak in een apart document behandeld, vooral voor 16bit- en 32bit-kernen;
- een gedetailleerde uitleg van complexe periferie: USB, grafische interfaces en Ethernet-randapparatuur worden meestal in aparte documenten behandeld, omdat de omvang van de datasheet van de microcontroller daardoor kan verdubbelen;
- errata: eventuele fouten in de documentatie, of werkwijzen om fouten in de silicium-wafer op te lossen die niet door de fabrikant worden verholpen, worden hier behandeld.

Welke andere microcontroller-documentatie is beschikbaar?

Naast de datasheet worden gewoonlijk de volgende documenten beschikbaar gesteld:

- "Family Reference Manual": terwijl de datasheet je precies vertelt wat één specifieke microcontroller kan, geven dergelijke documenten een overzicht op hoger niveau van wat alle microcontrollers van een bepaalde controllerfamilie kunnen;
- Application notes: deze gaan veel dieper in op het gebruik van specifieke onderdelen, of een groep onderdelen, om een toepassing of interface te realiseren. De datasheet kan uitleggen hoe de CAN-periferie (Controller Area Network) gebruikt moet worden, maar de application note legt in detail uit hoe ze ingezet worden als onderdeel van een CAN-netwerk, met richtlijnen voor softwareprotocollen op hoger niveau en de keuze van geschikte transceivers;
- specificatie van de programmering: voor programmering in massaproductie-omgevingen worden hier in detail de vereiste spanningen en timingdiagrammen gegeven, en het protocol dat gebruikt wordt voor de programmeerinterface.

Hoe kun je een grote hoeveelheid documentatie in je opnemen?

Helaas is er geen manier om snel alle gegevens van een microcontroller en al zijn hulpmiddelen in je op te nemen. Voor beginners is het waarschijnlijk het beste om de datasheet te lezen in combinatie met een boek of artikel over de microcontroller die je gaat gebruiken. Elektor heeft niet alleen boeken voor wie geïnteresseerd is in PIC-microcontrollers; er zijn ook beginnersboeken voor de STM32 [4] en ARM-gebaseerde microcontrollers [5] in het algemeen. Of waarom probeer je niet gewoon een eenvoudig MSP430-project [6]? De combinatie van praktische voorbeelden in combinatie met twee soorten documentatie (een boek/artikel en de datasheet) zal helpen.

Als je al enigszins gevorderd bent, kun je je het beste richten op de hoofdstukken over de processorkern, de klokstructuur en het reset-blok, gevolgd door de periferie die je wilt gebruiken. Daarna moet je je vertrouwd maken met de documentatie van de compiler-toolchain. Maak ook veel gebruik van softwarebibliotheken en voorbeelden om meer begrip te krijgen, en stel vragen op fora als je meer hulp nodig hebt.

Datasheets kunnen lastig zijn om te lezen en te begrijpen, en vormen niet de meest boeiende vorm van literatuur. Ze zijn echter (meestal) nauwkeurig en vormen een technische beschrijving van de functionaliteit (de errata kunnen worden gebruikt om in te schatten in hoeverre je de datasheet kunt vertrouwen). Geef niet te snel op en na verloop van tijd zul je een goed begrip ontwikkelen van hoe datasheets zijn opgebouwd en geformuleerd. ◀

vertaling: Hans Adams — 230286-03

Vragen of opmerkingen?

Hebt u technische vragen of opmerkingen naar aanleiding van dit artikel? Stuur een e-mail naar de auteur via stuart.cording@elektor.com.



Gerelateerde producten

- **T. Hanna, *Microcontroller Basics with PIC*, Elektor 2020**
www.elektor.nl/19188
- **A. Pratt, *Programming the Finite State Machine*, Elektor 2020**
www.elektor.nl/19327

WEBLINKS

- [1] Microcontroller-documentatie serie: <http://www.elektormagazine.nl/magazine/elektor-295/61525>
- [2] Microchip Technologies PIC16F18877 Datasheet: <https://microchip.com/en-us/product/PIC16F18877>
- [3] De D-flipflop: https://en.wikipedia.org/wiki/Flip-flop_%28electronics%29#D_flip-flop
- [4] Beginnersboeken voor de STM32: <https://elektor.com/programming-with-stm32-nucleo-boards-e-book>
- [5] ARM-gebaseerde microcontrollers: <https://elektor.com/embedded-in-embedded>
- [6] Simpel MSP430-project: <https://elektormagazine.com/labs/elektorpost-no-1-led-earring>



Farnell
AN AVNET COMPANY

SOFTWAREOPLOSSINGEN VOOR TESTEN EN METEN

Wij leveren nu je digitale softwareoplossing bij je hardware.

Kies je T&M software en ontvang deze binnen 72 uur na aankoop per e-mail.

