

Analóg és digitális áramkörök tesztelése és programozása peremfigyeléssel – 10.

Flashmemória és PLD programozása peremfigyeléssel

Dr. Kohut József főiskolai docens, Molnár Zsolt főiskolai tanársegéd

Budapesti Műszaki Főiskola Kandó Kálmán Villamosmérnöki Főiskolai Kar, Műszertechnikai és Automatizálási Intézet

Cikksorozatunk előző részében azoknak az áramköröknek a tesztelését tárgyaltuk, amelyek egy peremfigyeléssel rendelkező rendszer részei, de önmagukban nem tartalmaznak peremfigyelésre alkalmas integrált áramköröket. A mostani részben a programozható áramköri elemek – flashmemóriák és PLD-k – programozásának módszerébe tekintünk be.

Sok, napjainkban gyártott elektronika tartalmaz flashmemóriát vagy valamilyen programozható logikát (CPLD-t vagy FPGA-t). Ezeket az eszközöket a működésükhöz fel kell tölteni tartalommal: például a flashben el kell helyezni a beágyazott processzor számára a programot, vagy az FPGA számára a konfigurációs bitmintát, illetve a CPLD-t fel kell konfigurálni. Ráadásul gyakran előfordul, hogy a hardver változtatása nélkül egy szoftverfrissítés érdekében ismét el kell végezni ezt a műveletet. Ezt hagyományos módon úgy lehetett megtenni, hogy az adott eszközt kiemelték az áramkörből (foglatból), és külső programozóval átprogramozták. Ugyanakkor ezekben az összetett elektronikákban többnyire ki van építve a peremfigyeléses hurok, amelyen keresztül a tesztelésen kívül a fenti tevékenységek – a programozandó eszköz kiemelése nélkül – is elvégezhetőek. Fontos megemlíteni, hogy beágyazott peremfigyelés-vezérlővel akár a táv-szoftverfrissítés is lehetséges. A továbbiakban a flashmemória programozásáról lesz szó, de lényegét tekintve a PLD-programozás is hasonlóan történhet.

A flashmemória programozása és a hozzá tartozó műveletek (például törlés, visszaolvasás, adott területek levédése) igen összetett szekvenciákat igényelnek. Egy peremfigyeléssel nem rendelkező típus (AMD Am29F010) parancslistáját az alábbi táblázatban láthatjuk (1. táblázat).

Látható, hogy az adott típusnál változatos ciklusszámú (1...6), esetenként bonyolult szekvenciák valósítják meg az egyes paran-

csokat. Ezekre a szekvenciákra azért van szükség, hogy a memória véletlen írását vagy törlését elkerüljék. Például a tartalom teljes törlése hatciklusú művelet: két, úgynevezett feloldó (unlock) ciklussal kezdődik, majd ezt követi egy előkészítő (setup) ciklus, aztán ismét két feloldó ciklus jön, végül a teljes törlésparancs kiadásának ciklusa. Ugyanakkor a leggyakoribb flashmemória művelet, az olvasás, csupán egyciklusú. A törlés és a beprogramozás tehát meglehetősen sok időt vesz igénybe a sok részművelet miatt. Ezzel kapcsolatban megjegyzendő, hogy az ilyen típusú memóriáknál – a RAM-okkal szemben – maga a belső írási művelet is jóval hosszabb, mint az olvasás (1...5 ms).

A flashmemóriák peremfigyeléssel történő programozásához a peremfigyeléses hurokba kapcsolt flashmemória TAP-interfészén, illetve a peremfigyeléses funkciókon keresztül a memória összes funkcionális kivezetéséhez hozzá kell férni. A beírásához szükséges vezérlő- és adatjelek a memória cím- és adatkivezetési helyett a peremfigyeléses hurkon keresztül lépnek be. A beléptetés után ugyanúgy, mintha „kívülről” érkeztek volna a jelek, egy parancs hatására megtörténik a beírás.

A beírás egy beíró jel hatására következik be, amely létrehozható szoftveres vagy hardveres úton. A szoftveres megvalósításnál – mivel a beírás H-L-H vagy L-H-L átmenetet igényel – a teljes peremfigyelési hurkot háromszor kell feltölteni (1. ábra). Természetesen a hurok feltöltési idejének csökkentése érdekében a hurokban lévő, az éppen programozott memórián kívüli elemek áthidaló

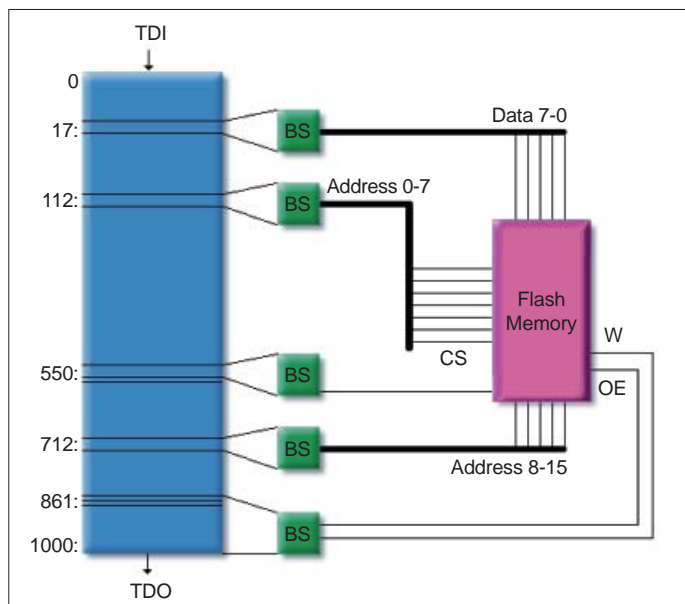
A parancs szekvenciája	A művelet ciklusszáma	Buszciklusok											
		1		2		3		4		5		6	
		Cím	Adat	Cím	Adat	Cím	Adat	Cím	Adat	Cím	Adat	Cím	Adat
Olvasás	1	RA	RD										
Alaphelyzetbe állítás (autoselect után)	1	XXXX	F0										
Autoselect	Gyártóazonosító	4						XX00	01				
	Eszközazonosító	4	5555	AA	2AAA	55	5555	90	XX01	20			
	Védettség lekérdezés	4							(SA)X02	00/01			
Programozás	4	5555	AA	2AAA	55	5555	A0	PA	PD				
Teljes törlés	6	5555	AA	2AAA	55	5555	80	5555	AA	2AAA	55	5555	10
Szektor törlése	6	5555	AA	2AAA	55	5555	80	5555	AA	2AAA	55	SA	30

Magyarázat

- minden érték hexadecimálisan értendő
- RD: az olvasott adat
- az Autoselect parancs 4. ciklusában olvasás van

- RA: az olvasandó cella címe
- SA: a védett terület (szektor) azonosítója fél bájtton (0...7)
- PA: a programozandó cella címe

1. táblázat Az Am29F010 típusú memória parancsai



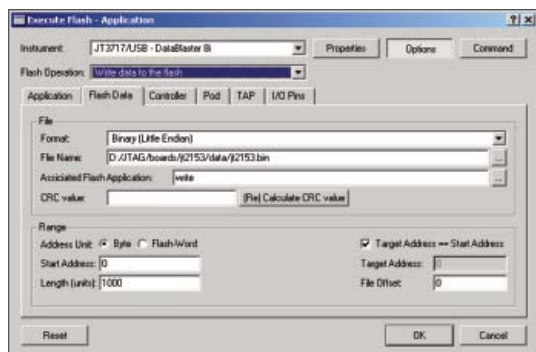
1. ábra A memória beíró impulzusának létrehozása szoftveres úton

(Bypass) módba tehetők. Ha a beíró jelet hardveresen hozzák létre, akkor optimális esetben akár a harmadára csökken a programozási idő. A JTAG Technologies peremfigyelés-vezérlői támogatják ezt a funkciót, és képesek létrehozni ezt a vezérlőjelet (AW jel), amely a TAP többi jelével együtt elérhető a vezérlők csatlakozóján.

A beírandó memóriatartalom hibátlanságáról, annak beírásáról és visszaellenőrzéséről a vezérlő szoftver gondoskodik. Általában a kezelő szoftverben további funkciók is rendelkezésre állnak, mint például: kezdőcímetolás, védett területek lekérdezése, tartalom (részleges vagy teljes) törlése stb.

Egy beírási folyamat tehát alapesetben a memóriatartalmat hordozó fájl beolvasásából, a flashmemória beprogramozásából, illetve a visszaellenőrzésből áll. Ugyanakkor lehetőség van egy már beprogramozott memória tartalmának visszaolvasására és fájlba mentésére is. A beírás folyamatát a JTAG Technologies peremfigyelés vezérlőjével a következő ábra szemlélteti (2. ábra).

A memóriatartalmat tartalmazó fájl beolvasásra kerül a peremfigyelés vezérlő belső, átmeneti memóriájába (Image memory), majd innen jut kiírásra a flashbe. A ProVision programban a következő ablakban (Execute Flash) lehet beállítani a flashmemóriával kapcsolatos műveleteket (3. ábra).



3. ábra A ProVision flashprogramozás funkciójához tartozó konfiguráló ablak

Az egyes mezőkben a következő beállítások láthatóak:

- Instrument: a peremfigyelés vezérlő típusának beállítása,
- Flash Operation: aktuális flashművelet (pl. memóriatartalmat hordozó fájl betöltése, memória írása, olvasása, tartalom ellenőrzése stb.) kiválasztása,
- Format: a memóriatartalmat hordozó fájl formátuma,
- File Name: a fájl elérési útja és neve,



2. ábra A flashbe írás folyamata

- Associated Flash Operation: a fájlművelet beállítása,
- CRC Value: a fájl tartalom ellenőrző összege (CRC: Calculated Redundancy Check),
- Address Unit: annak kiválasztása, hogy a következőkben beállított címek bájt- vagy szócímekeket jelentenek-e,
- Start Address: a kezdőcím beállítása. Több fájl betöltése is lehetséges, természetesen ilyenkor a kezdőcím beállított értékétől függően a már feltöltött terület egy részét vagy egészét felül lehet írni,
- Length (units): a memóriaművelet tartományát lehet megadni az előzőekben meghatározott egységekben (bájt/szó),
- Target Address: Cél kezdőcímeinek beállítása; kiválasztható, hogy a célcím megegyezzen a kezdőcímmel (Target Address == Start Address),
- File Offset: fájlbetöltésnél a fájl tartalom egy részét átugorva lehet kezdeni a műveletet.

A programozható logikai eszközök (CPLD-k és FPGA-k) általában felkonfigurálás nélkül kerülnek beültetésre az áramkörökbe. Az FPGA-kat – mivel a konfigurációt a csipen megvalósított felejtő memóriában tárolják – a tápfeszültség bekapcsolás után minden esetben fel kell konfigurálni. Ez történhet a hozzákapcsolt flashmemóriából (Boot Flash) automatikusan, vagy a konfigurációs bitmintát a beagyazott peremfigyelés vezérlő töltheti be. A CPLD-k nem felejtő memóriáját – a peremfigyeléses hurkot felhasználva – feltölthetik gyártáskor, vagy az FPGA-khoz hasonlóan át lehet azokat programozni a beagyazott peremfigyelés vezérlővel. A programozható logikai eszközök (és így a szükséges felkonfigurálások) számának növekedése ösztönözte a fejlesztőket és gyártókat, hogy hatékony módszert dolgozzanak ki erre a célra. Az IEEE1149.1 szabványra épülve 2000-ben az IEEE megalkotta az IEEE1532-es szabványt, amely a rendszerben konfigurálás (In-System Configuration, ISC) eljárását szabványosítja. Ma már az IEEE1532 szabvánnyal kompatibilis eszközöket minden jelentős PLD-gyártó gyárt (pl. Xilinx, Altera, Lattice).

Az 1532-es szabványnak megfelelő eszközök a 1149.1-kompatibilis TAP-ot felhasználva konfigurálhatóak, néhány járulékos belső regiszter, belső jel és parancs felhasználásával.

Ezek a kiegészítő regiszterek:

- ISC_Default: általában az áthidaló (Bypass) regisztert használják erre a célra, de a szabvány szerint megvalósítható külön regiszterként is. Ha nincs konfigurálással kapcsolatos regiszterművelet, akkor ez a regiszter kapcsolódik a peremfigyeléses hurokba,
- ISC_Config: a programozás opcionális paramétereit tárolja,
- ISC_PData: a programozási címet vagy adatot tároló regiszter,
- ISC_RData: visszaolvasási regiszter,
- ISC_Status: a két járulékos belső jel, az ISC_Enable és az ISC_Done jelek állapotáról ad információt.

Az ISC_Enable belső jel engedélyezi a rendszerben konfigurálást, az ISC_Done pedig jelzi a konfigurálási művelet sikeres befejezését.

Ezzel áttekintettük a peremfigyeléses tesztelés és programozás technikáját. A következő részekben a gyártásközi és végellenőrzés, valamint az üzemeltetés alatti alkalmazásokat tárgyaljuk.