

Anàlisi de la interconnexió de dispositius lògics programables mitjançant Ethernet

Marko Peshevski

marko.peshevski@estudiant.upc.edu

Resum

En aquest treball s'estudia com fer funcionar Ethernet des d'una placa amb FPGA (de l'anglès Field Programmable Gate Array). Aquests són dispositius electrònics que permeten reprogramar la lògica que contenen dins per tal que aquesta es comporti com l'usuari necessiti per l'aplicació en qüestió. Una de les maneres que existeixen de fer funcionar una FPGA és fent que aquesta es comporti com si fos un microcontrolador incrustat (MicroBlaze, del fabricant Xilinx, en aquest cas). A més de fer funcionar Ethernet s'implementa en llenguatge C una pila de protocols bàsica capaç de respondre a peticions d'eco ICMP (Internet Control Message Protocol). Per altra banda, aquestes peticions d'eco també es resolen utilitzant la llibreria LwIP (Lightweight Internet Protocol) àmpliament coneguda i utilitzada en el món de les aplicacions incrustades. Es comparen aquestes dues implementacions, resultant que la implementació bàsica és més ràpida en respondre a una petició d'eco.

1. Introducció

En aquest treball s'estudien els dispositius coneguts com FPGA, de l'anglès Field Programmable Gate Array. A grans trets, són dispositius que permeten reprogramar les connexions entre els blocs lògics a l'interior dels mateixos, per aconseguir que funcionin tal com l'usuari necessita per l'aplicació que estigui dissenyant. El principal avantatge que ofereixen aquests dispositius respecte a d'altres dispositius similars i d'altres mètodes per implementar funcions lògiques és la seva gran integració (solen ser circuits integrats amb un nombre d'elements lògics des de desenes de milers fins a milions), i la seva reprogramabilitat, a diferència dels circuits integrats i sistemes sobre xip dedicats.

Per altra banda, en aquest mateix treball s'analitza i estudia la connectivitat Ethernet. Aquest mètode de connexió és àmpliament conegut i àmpliament utilitzat, tant a nivells industrials com a nivells d'electrònica de consum. És, per tant, de gran interès conèixer com funciona, i quines són les seves possibilitats i limitacions. Per sobre d'Ethernet, que només correspon a les capes física i de control d'accés al medi, generalment s'hi poden trobar altes protocols, com ara: ARP, TCP/IP, Token Ring, Token Bus, etcètera. Aquests protocols són els que realment treballen amb les dades que es volen intercanviar entre els dos (o més) punts a la xarxa que estiguin comunicats.

En la present obra s'estudia i analitza com es poden unir tots dos mons, i quins són els avantatges i inconvenients. De totes formes, però, amb Ethernet sola només es pot crear una xarxa local. Per tant, existeixen moltes implementacions de la pila de protocols necessaris per connectar-se a Internet (TCP/IP) sobre moltes arquitectures diferents. En aquest document s'empra una de les implementacions més conegudes, LwIP. Generalment aquesta implementació s'utilitza en sistemes incrustats, amb microprocessadors restringits en espai de memòria. Aquesta implementació es compara contra una implementació molt més bàsica, que respon només a algun dels protocols més bàsics d'una possible pila de protocols per establir una connexió entre equips connectats en una xarxa.

2. FPGA

Aquests dispositius, el nom dels quals traduït literalment és: formació de portes (lògiques) programables al camp (*in situ*); són un tipus de dispositius electrònics que permeten la generació de funcions lògiques, i aplicacions més complexes, mitjançant la reprogramació de l'estat dels seus blocs lògics i l'estat de les interconnexions entre aquests. Una imatge qualitativa de l'estructura interna d'una FPGA podria ser la de la Figura 1.

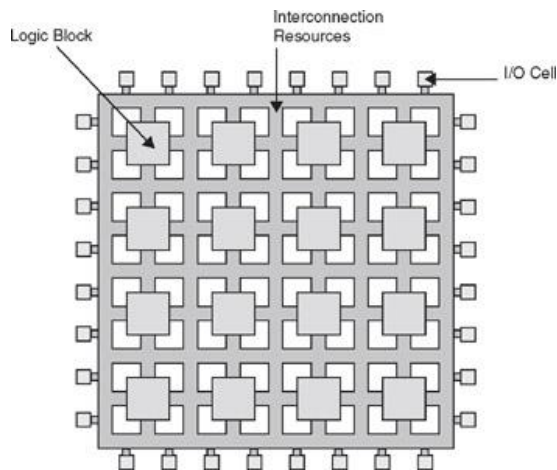


Fig. 1. Estructura interna de la FPGA altament simplificada

Un bloc lògic dels que es mencionen a la figura podria tenir un funcionament similar al descrit a la Figura 2. Generalment solen ser més complets i incorporar LUTs (Look Up Tables) amb més entrades. És important que els blocs lògics incorporin biestables a la sortida d'aquests, per poder aconseguir que tota la lògica funcioni sincronitzada.

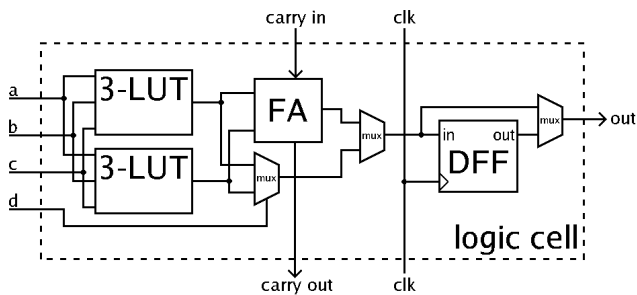


Fig. 2. Cel·la lògica simplificada d'una FPGA

Els recursos d'interconnexió de la FPGA són els que es fan servir per connectar els blocs entre si de la manera desitjada. La Figura 3 és una representació simplificada de com podria ser un dels interruptors d'interconnexió de les FPGA.

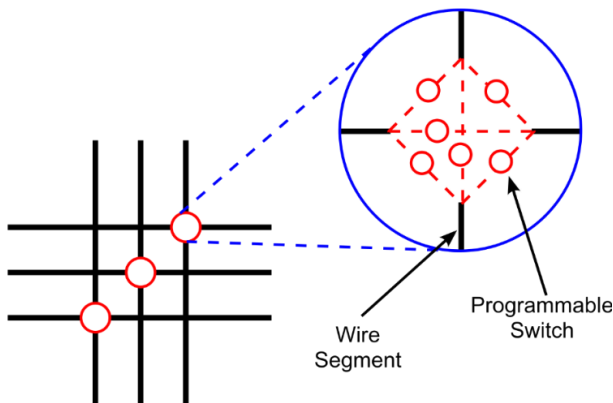


Fig. 3. Interruptor d'interconnexió de la matriu de la FPGA

Per últim a les FPGA sol haver-hi blocs que gestionen l'entrada i sortida de senyals d'aquestes cap al món exterior. Aquests blocs d'entrada/sortida solen incorporar electrònica per poder controlar-los com entrada, sortida i alta impedància (mitjançant buffers tri-estat). La Figura 4 és una representació simplificada de com podria ser la electrònica d'un bloc d'entrada/sortida.

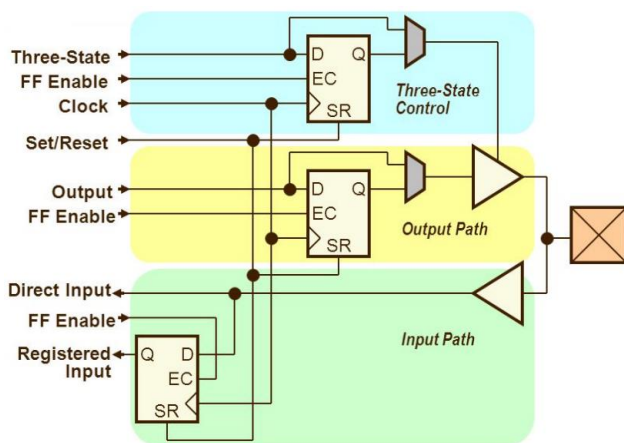


Fig. 4. Bloc d'entrada/sortida d'una FPGA

Com que les FPGA són dispositius tan densos a nivell d'electrònica, es poden fer aplicacions relativament complexes amb aquestes. Quan un programa de FPGA fa que aquesta es comporti d'una determinada manera, com si fos una altra peça de hardware, se sol anomenar IP core (nucli de propietat intel·lectual). Uns dels IP cores més

utilitzats són els que fan que la FPGA es comporti com si tingués un microcontrolador incrustat (anomenats soft-cores). El fabricant Xilinx anomena el seu soft-core MicroBlaze. És un microcontrolador que s'assembla a un dels més moderns ARM.

3. Ethernet

Ethernet és un protocol que se situa als nivells 1 i 2 del Model OSI que defineix un funcionament recomanat per una pila de protocols per interconnectar equips a través de la xarxa. Es pot trobar una representació del Model OSI a la Figura 5.

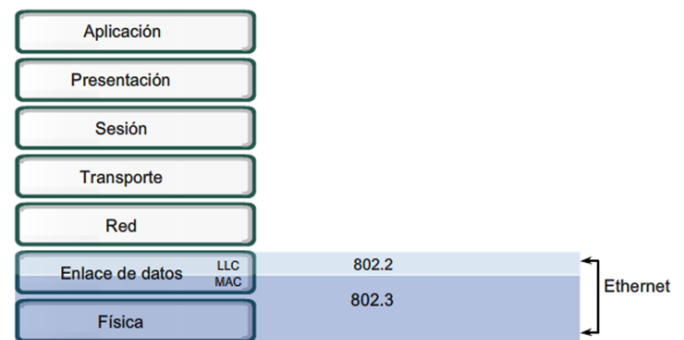


Fig. 5. Model OSI, destacant Ethernet

Per sobre d'Ethernet s'hi munten altres protocols del model que tracten amb les dades, gestionen la xarxa, etcètera. En una xarxa Ethernet és molt important que cada dispositiu tingui una adreça diferent dels altres, per evitar col·lisions de paquets. Aquestes adreces són de 48 bits (6 bytes) i solen dividir-se en meitats de 3 bytes, dels quals els primers 3 corresponen al fabricant del dispositiu, i els darrers 3 identifiquen aquella família de dispositius.

Un diagrama de blocs que descriu el funcionament d'Ethernet a la placa utilitzada es pot veure a la Figura 6.

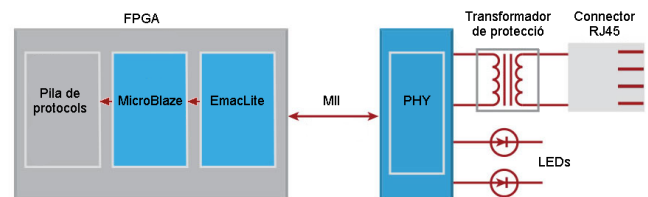


Fig. 6. Diagrama de blocs del funcionament d'Ethernet a la placa utilitzada

En aquesta figura s'hi poden identificar 3 parts clarament diferenciades. Per poder establir una connexió Ethernet es necessiten totes tres. Una d'elles és el connector físic per connectar cables Ethernet (estàndard RJ45), amb el transformador de protecció. La segona part és el PHY. Normalment sol ser un circuit integrat dissenyat específicament per aquesta funció. Aquest xip només es dedica a decodificar els senyals que li arriben a través dels cables i traduir-los per tal que el dispositiu següent (en aquest cas la FPGA i MicroBlaze) pugui entendre i tractar les trames d'Ethernet.

4. Protocols utilitzats

Els protocols que s'utilitzen en aquest treball, per sobre d'Ethernet, són:

- IPv4
- ARP
- ICMP

IPv4 és el protocol bàsic que s'utilitza per l'intercanvi de dades tant dins d'una xarxa local com a la xarxa de xarxes. Aquest és un protocol extens que permet intercanvi de dades dintre d'una xarxa on tots els equips connectats estan adreçats amb una adreça IP, diferent de l'adreça d'Ethernet (MAC), que permet identificar-los independentment del dispositiu, i permet organització lògica dels equips, mitjançant la formació de xarxes i subxarxes. A la Figura 7 es pot veure una representació dels camps d'un paquet IPv4.

Paquet IPv4					
Versió 4 bits	Llargària de la capçalera 4 bits	Serveis Diferenciats 6 bits	N C E 2 bits	Llargària total del paquet 16 bits	
Identificació 16 bits				Flags 3 bits	Número de fragment 13 bits
Temps de vida 8 bits		Protocol 8 bits		Suma de verificació de la capçalera 16 bits	
Adreça IP d'origen 32 bits					
Adreça IP de destí 32 bits					
Dades					

Fig. 7. Camps d'un paquet IPv4

ARP és un protocol que serveix per resoldre les adreces físiques (MAC) dels dispositius connectats a una xarxa. Per utilitzar aquest protocol, l'equip que necessiti saber la direcció física d'un altre pot emetre un paquet Ethernet *broadcast* (dirigit a tots els dispositius d'aquella xarxa física), al qual només contestarà aquell dispositiu amb l'adreça lògica que estigui sent cridat. A la Figura 8 es pot veure una representació dels camps d'un paquet ARP.

Paquet ARP	
Tipus de medi 16 bits	
Identificació 16 bits	
Llargària adreça física 8 bits	Llargària adreça lògica 8 bits
Operació que es realitza 16 bits	
Adreça física d'origen 48 bits (6 bytes)	
Adreça lògica d'origen 32 bits (4 bytes)	
Adreça física de destí 48 bits (6 bytes)	
Adreça lògica de destí 32 bits (4 bytes)	

Fig. 8. Camps d'un paquet ARP

ICMP és un protocol que serveix per regular el tràfic a la xarxa. Això es refereix a, per exemple, notificar a l'originador d'una transmissió de la pèrdua del seu paquet, o que el paquet no ha pogut arribar al destinatari en el nombre de salts (temps de vida) establert. Entre d'altres, aquest protocol permet fer peticions i respostes de paquets anomenats eco. Això permet comprovar l'estat de la xarxa, i en quin temps es poden enviar i rebre paquets d'un lloc a un altre. A la Figura 9 es pot veure una representació dels camps d'un paquet ICMP.

Paquet ICMP		
Tipus 8 bits	Codi 8 bits	Suma de verificació del paquet 16 bits
Identificador 16 bits		Número de seqüència 16 bits
Dades		

Fig. 9. Camps d'un paquet ICMP

5. Implementacions pràctiques

Com s'ha dit anteriorment, en aquest treball s'estudien dues implementacions pràctiques. Una d'aquestes està basada en una popular llibreria de pila de protocols d'Internet, anomenada LwIP, mentre que l'altra és una implementació feta per l'autor. La llibreria LwIP només requereix d'una interfície de xarxa (Ethernet), un temporitzador i un controlador d'interrupcions per funcionar, però necessita un espai de memòria considerable (40 kilobytes de memòria de programa). Aquesta llibreria, però, és fàcil a l'hora de fer-la servir en una aplicació. Un cop inicialitzats els perifèrics que aquesta necessita, i la llibreria en sí, pràcticament *funciona sola*.

Per altra banda, la implementació feta per l'autor, encara que més bàsica, només necessita la interfície Ethernet i un controlador d'interrupcions per funcionar. A més d'això, és un codi més curt, aproximadament ocupa uns 13 kilobytes, incloent missatges per UART i altres comoditats no imprescindibles.

6. Resultats experimentals

Un cop fetes les implementacions pràctiques totes dues es poden provar per veure quina és més ràpida a l'hora de contestar una petició d'eco. Això es fa creant un script de consola de Linux que executa cert nombre de peticions d'eco amb diferents llargàries de paraula, per veure com es comporten totes dues implementacions.

Un resultat destacable es pot veure a la Figura 10, on es pot comprovar que la implementació bàsica feta per l'autor (línies inferiors), és més ràpida (té un RTT, round-trip time, més baix) que la implementació utilitzant LwIP.

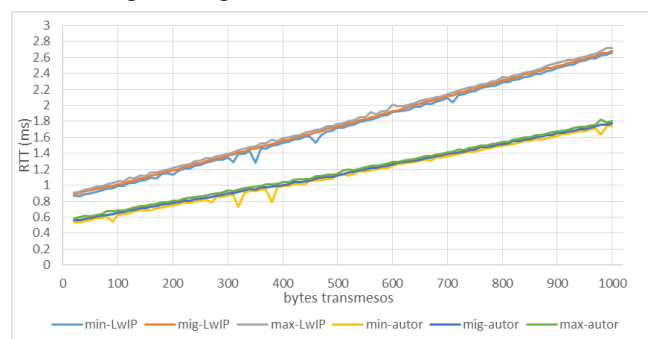


Fig. 10. Resultats experimentals de comparació entre la implementació de l'autor i LwIP

Durant la programació del codi s'observa que segurament el coll d'ampolla que fa que les aplicacions d'aquest estil vagin més lentes és, segurament, una programació ineficient dels algorismes per generar sumes de verificació dels paquets. Això també es pot comprovar a la Figura 11, on les línies de baix corresponen a la mateixa implementació que en la figura anterior, les línies del mig corresponen a la implementació amb LwIP, i les línies superiors corresponen a una implementació més lenta (traducció directa de l'algorisme) de la suma de verificació dels paquets ICMP.

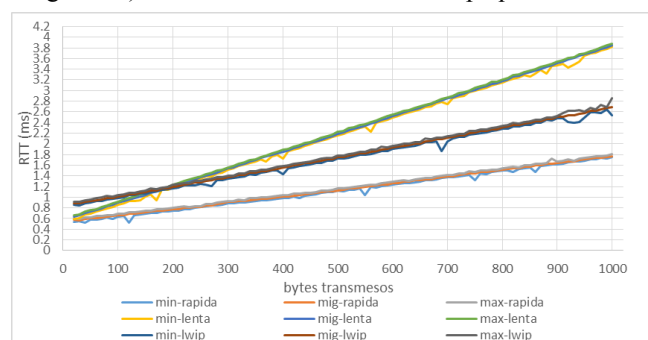


Fig. 11. Resultats experimentals de comparació entre diferents mètodes de suma de verificació

Degut a limitacions d'espai de memòria durant el desenvolupament de les implementacions pràctiques es decideix utilitzar una memòria RAM LPDDR externa que duu incorporada la placa amb FPGA utilitzada. Això dóna

un espai de memòria per programa pràcticament il·limitat per aquesta aplicació (64 MB), però se sospita que amb el desavantatge de ser més lent. Això es pot comprovar veient la Figura 12, on es mostra la mateixa implementació feta per l'autor executant-se des de la memòria RAM i executant-se des de la memòria interna del MicroBlaze. En aquesta comparativa no s'hi pot incloure LwIP degut a què aquesta llibreria no entra a l'espai de memòria màxim disponible a MicroBlaze (32 kilobytes).

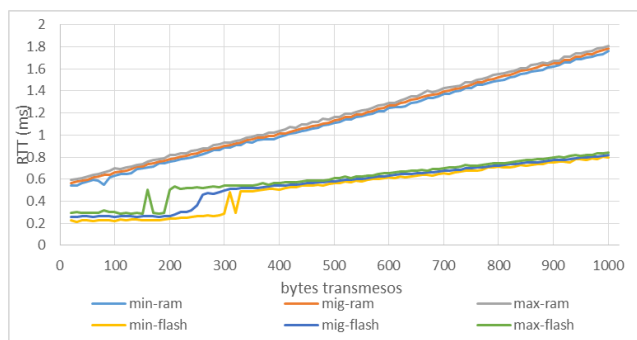


Fig. 12. Resultats experimentals de comparació entre la implementació de l'autor i LwIP

En aquesta figura es pot veure que la execució de codi des de RAM externa (línies superiors) és notablement més lenta que la execució de codi des de la memòria interna de MicroBlaze.

7. Conclusions

Com s'ha pogut comprovar amb les proves dutes a terme, sembla ser que cada manera d'implementar la pila de protocols té els seus avantatges i inconvenients. Sembla que LwIP seria una millor opció amb un hardware més ampli, amb més memòria i un processador més capaç. Un altre gran avantatge que té aquesta llibreria és que suporta molts més protocols (TCP, UDP, DHCP, etcètera) que els de la pila feta per l'autor. Tots aquests protocols serien imprescindibles per una aplicació connectada a Internet (per exemple, un servidor web). En canvi desenvolupar una pila de protocols pròpia basada en el protocol IP sembla una bona idea si s'ha de fer servir en alguna aplicació *tancada al món exterior*, en el sentit de sense connectivitat a Internet. El gran avantatge de la pila de protocol feta per l'autor és que pot respondre a peticions més ràpid.