



**Escola Politècnica Superior
d'Enginyeria de Vilanova i la Geltrú**

UNIVERSITAT POLITÈCNICA DE CATALUNYA

TREBALL FINAL DE MÀSTER

TÍTOL: Anàlisi de la interconnexió de dispositius lògics programables mitjançant Ethernet

AUTOR: Peshevski, Marko

DATA DE PRESENTACIÓ: Febrer, 2017

COGNOMS: Peshevski

NOM: Marko

TITULACIÓ: Màster Universitari en Enginyeria de Sistemes Automàtics i Electrònica Industrial (MUESAEI)

PLA:

DIRECTOR: Mariano López García

DEPARTAMENT: EEL - Departament d'Enginyeria Electrònica

QUALIFICACIÓ DEL TFM

TRIBUNAL

PRESIDENT

SECRETARI

VOCAL

DATA DE LECTURA:

Aquest Projecte té en compte aspectes mediambientals: ☐ Sí ☐ No

RESUM

Paraules clau (màxim 10):

ABSTRACT

Keywords (10 maximum):

SUMARI

1. INTRODUCCIÓ.....	8
1.1. Objectius	8
2. FPGA.....	9
2.1. Estructura.....	9
2.2. Nuclis de propietat intel·lectual	11
2.3. MicroBlaze.....	12
2.3.1. Generació de Hardware	12
2.3.2. Programació de Software	14
3. ETHERNET	15
3.1. Pila de protocol per capes	15
3.2. Funcionament d'Ethernet a la placa utilitzada	17
3.3. Protocols utilitzats	18
4. IMPLEMENTACIÓ PRÀCTICA.....	23
4.1. LwIP	23
4.2. Pila programada per l'autor	23
5. RESULTATS EXPERIMENTALS	0
5.1. Mètode d'estudi	0
5.2. Resultats estadístics	0
5.3. Comparació entre les piles	0
6. CONCLUSIONS I TREBALL FUTUR	0
6.1. Conclusió	0
6.2. Treball futur.....	0
7. BIBLIOGRAFIA I ANNEXES	0

SUMARI DE FIGURES

FIGURA 1. ESTRUCTURA DE LA FPGA, ALTAMENT SIMPLIFICADA	9
FIGURA 2. CEL·LA LÒGICA INDIVIDUAL SIMPLIFICADA D'UNA FPGA	9
FIGURA 3. REPRESENTACIÓ SIMPLIFICADA D'UN DELS INTERRUPTORS D'INTERCONNEXIÓ D'UNA FPGA	10
FIGURA 4. REPRESENTACIÓ SIMPLIFICADA D'UN DELS BLOCS D'ENTRADA/SORTIDA DE LA FPGA	10
FIGURA 5. FOTO DE LA PLACA UTILITZADA EN AQUEST TREBALL	11
FIGURA 6. EXTRACTE D'UNA LLISTA DE IP CORES DISPONIBLES PER L'USUARI	12
FIGURA 7. VISTA DE CONFIGURACIÓ DELS MÒDULS IP CORE TRIATS PEL PROJECTE EN CURS	13
FIGURA 8. EXTRACTE D'UNA LLISTA DE IP CORES DISPONIBLES PER L'USUARI	13
FIGURA 9. CAPTURA DE PANTALLA DEL PROGRAMARI SOFTWARE DEVELOPMENT KIT	14
FIGURA 10. REPRESENTACIÓ DEL MODEL OSI	15
FIGURA 11. DIFERENTS CATEGORIES DE CABLEJAT PER ETHERNET QUE ES PODEN TROBAR	16
FIGURA 12. REPRESENTACIÓ DE VÀRIES TOPOLOGIES DE XARXES POSSIBLES	16
FIGURA 13. DIAGRAMA QUE REPRESENTA EL FUNCIONAMENT D'ETHERNET A LA PLACA UTILITZADA	17
FIGURA 14. DIAGRAMA QUE REPRESENTA LA ESTRUCTURA D'UNA TRAMA ETHERNET	19
FIGURA 15. DIAGRAMA QUE REPRESENTA LA ESTRUCTURA D'UN PAQUET IPV4	20
FIGURA 16. DIAGRAMA QUE REPRESENTA LA ESTRUCTURA D'UN PAQUET ARP	21
FIGURA 17. DIAGRAMA QUE REPRESENTA LA ESTRUCTURA D'UN PAQUET ARP	22

1. INTRODUCCIÓ

En aquest treball s'estudien els dispositius coneguts com FPGA, de l'anglès Field Programmable Gate Array. Aquests dispositius han crescut en popularitat al llarg de les últimes dècades degut a què cada cop s'han fet més accessibles i han incorporat major nombre d'elements lògics. A grans trets, són dispositius que permeten reprogramar les connexions entre els *blocs lògics* a l'interior dels mateixos, per aconseguir des de funcions lògiques senzilles, fins a aplicacions relativament complexes que necessitin d'una elevada densitat a nivell d'electrònica. 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 molt densos, 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. Aquesta connectivitat ja té una llarga vida, existeix des de la dècada dels 1970. És àmpliament coneguda i àmpliament utilitzada, 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. Essent que TCP/IP és una pila de protocols de comunicació global i molt utilitzada, existeixen moltes implementacions de la mateixa 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, feta per l'autor, que permet intercanviar dades entre dos nodes en una xarxa ja existent, que respon només a algun dels protocols més bàsics d'una possible pila de protocols per establir una connexió entre dos equips connectats en una xarxa.

1.1. Objectius

Alguns dels objectius que s'han perseguit amb aquest treball són:

- Conèixer en profunditat el funcionament d'Ethernet a nivell de bit d'informació.
- Conèixer en profunditat alguns dels protocols més emprats a les comunicacions d'avui en dia (Internet).
- Dissenyar un sistema incrustat sobre una FPGA.
- Conèixer amb el màxim detall possible com funciona una implementació pràctica de la pila de protocols d'Internet per sistemes incrustats.
- Intentar desenvolupar una pila de protocol bàsica que permet respondre a peticions d'eco en una xarxa.
- Mesurar els temps que tarda una pila i l'altra en processar un paquet d'eco i comparar resultats.

2. FPGA

2.1. Estructura

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.

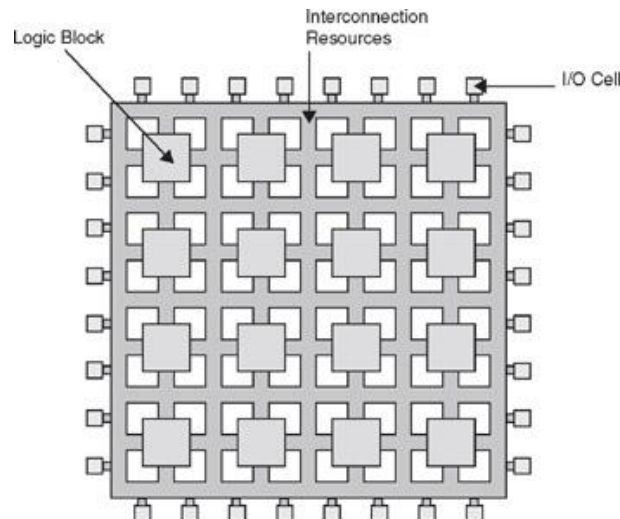


Figura 1. Estructura de la FPGA, altament simplificada

Els blocs lògics, coneguts també com cel·les lògiques són els elements capaços de realitzar funcions lògiques. Un exemple d'una cel·la es pot trobar a la Figura 2.

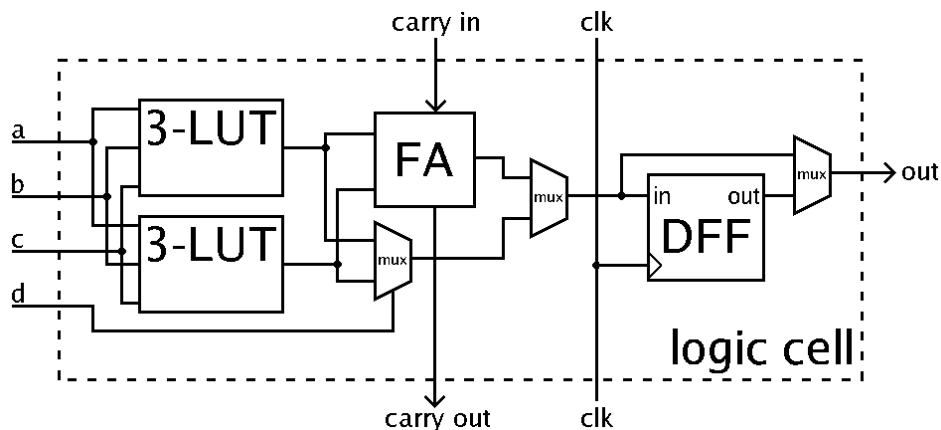


Figura 2. Cel·la lògica individual simplificada d'una FPGA

Segons es pot veure en aquesta Figura, cada cel·la d'una FPGA consisteix de 4 bits d'entrada (a, b, c i d), que permeten entrar informació a la cel·la a través de les taules d'entrada (LUT, Look Up Table), sigui del món exterior, com de la resta de cel·les. L'altra entrada (carry in), permetria encadenar cel·les per poder fer circuits sumadors més complexes utilitzant els sumadors complets (FA, Full-Adder de les cel·les), per exemple. En aquestes cel·les, el que se selecciona en el moment de la programació de la pròpia FPGA és l'estat dels multiplexors, per aconseguir que la cel·la es comporti d'una determinada manera. A la sortida hi ha un biestable de tipus D (DFF, D Flip-Flop), governat per un senyal de rellotge, que és, en general, global a totes o la

majoria de cel·les. Aquest biestable és molt important degut a què els circuits que s'hagin de generar a les FPGA normalment han de ser síncrons. Cal notar que una cel·la real sol ser més complexa, amb més entrades i probablement amb més elements lògics al seu interior.

Altres elements presents a les FPGA són els recursos d'interconnexió. Aquests permeten encaminar les connexions entre els blocs lògics i els blocs d'entrada/sortida. Aquests recursos d'interconnexió consisteixen d'interruptors programables que permeten seleccionar quins camins han de seguir les pistes d'interconnexió dels blocs lògics de la FPGA. Una representació altament simplificada es pot veure a la Figura 3.

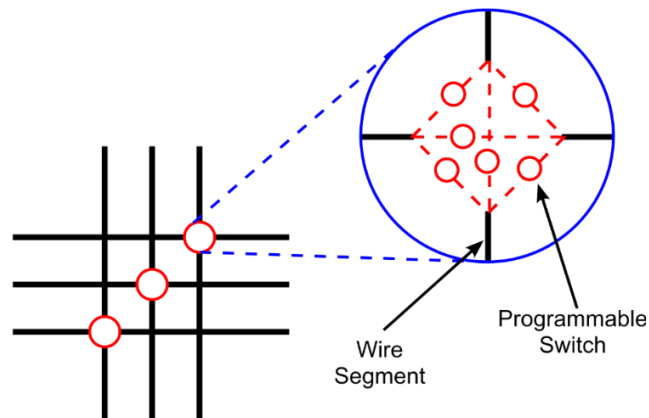


Figura 3. Representació simplificada d'un dels interruptors d'interconnexió d'una FPGA

Per últim, els elements que falta descriure d'una FPGA són els blocs d'entrada/sortida. Aquests blocs són trossos d'electrònica que permeten configurar les connexions de la lògica generada pels blocs lògics, i connectada pels recursos d'interconnexió, amb els pins que connecten la FPGA al món exterior. Generalment solen incorporar electrònica per poder fer que un pin en concret sigui entrada/sortida, o estigui en estat d'alta impedància (control tri-estat).

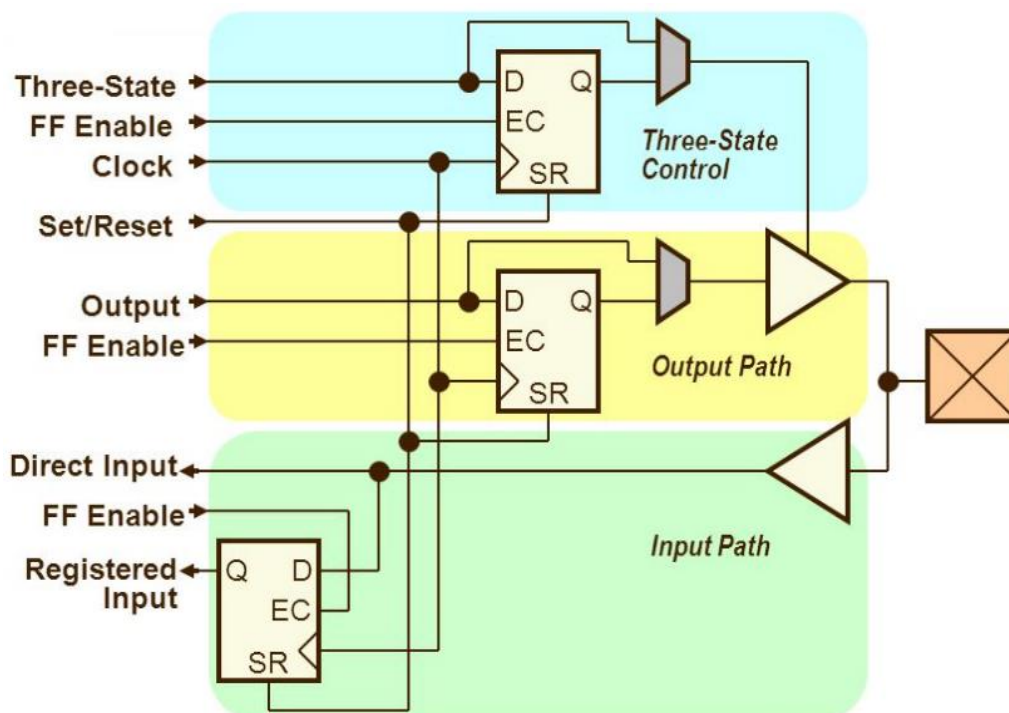


Figura 4. Representació simplificada d'un dels blocs d'entrada/sortida de la FPGA

2.2. Nuclis de propietat intel·lectual

Degut a l'àmplia disponibilitat de portes lògiques dintre d'una FPGA, aquestes es poden programar perquè es comportin com altres sistemes sencers. Per exemple, trossos de la lògica disponible a la FPGA es poden programar per comportar-se com un microcontrolador, amb uns perifèrics determinats, a triar per l'usuari segons necessitats de l'aplicació. La resta de la lògica disponible es podria fer servir per implementar, per exemple, funcions lògiques que necessiten ser executades molt ràpidament, com ara les d'una memòria d'accés aleatori disponible pel microcontrolador, o algun tipus de processat de senyal digital que s'executi en paral·lel amb el microcontrolador, de forma molt més ràpida.

Aquest tipus de programació aporta una flexibilitat molt gran, només limitada per la quantitat d'elements lògics disponibles. És per això que els grans fabricants de FPGA desenvolupen i permeten fer servir, de vegades sota llicència, els anomenats nuclis de propietat intel·lectual (de l'anglès Intellectual Property core). Els IP cores són implementacions en llenguatge de descripció de hardware (generalment VHDL o Verilog) d'algun dispositiu dins la lògica de la FPGA. Poden ser de diferents tipus: un controlador de memòria RAM DDR, un controlador d'accés al medi per Ethernet, un perifèric SPI, un microcontrolador sencer, etcètera. Quan aquests IP cores són implementats en llenguatge de descripció de hardware s'anomenen soft-cores. Més endavant, a la secció 4, es descriuran amb més detall els IP cores utilitzats en aquest treball.

Existeixen també versions permanents dels soft-cores, que són incrustats al silici de la pròpia FPGA, generalment en forma de microprocessador. Aquests últims s'anomenen hard-cores, i existeix una gran varietat dels mateixos. Generalment s'utilitzen de forma híbrida en conjunt amb la resta de la lògica de la FPGA, fent servir algun bus d'interconnexió entre totes dues parts. Els grans fabricants es decanten per un o un altre tipus. Per exemple, el fabricant Xilinx ofereix models de FPGA amb un microprocessador PowerPC incrustat, mentre que per altra banda Altera ofereix molts models amb un ARM incrustat.

Per aquest treball, degut a la disponibilitat al departament, s'utilitza una placa amb FPGA d'un dels principals fabricants del mercat, Xilinx. La placa en qüestió és la Avnet Spartan-6 LX9 MicroBoard. Aquesta placa duu una FPGA XC6SLX9 de Xilinx. Aquesta FPGA és un dels models bàsics del fabricant Xilinx, i l'usuari disposa entre d'altres, de 9152 blocs lògics amb taules d'ent i un màxim de 200 entrades/sortides. Aquesta placa duu incorporada connectivitat Ethernet fins a 100 Mb/s, que s'utilitzarà en aquest treball.



Figura 5. Foto de la placa utilitzada en aquest treball

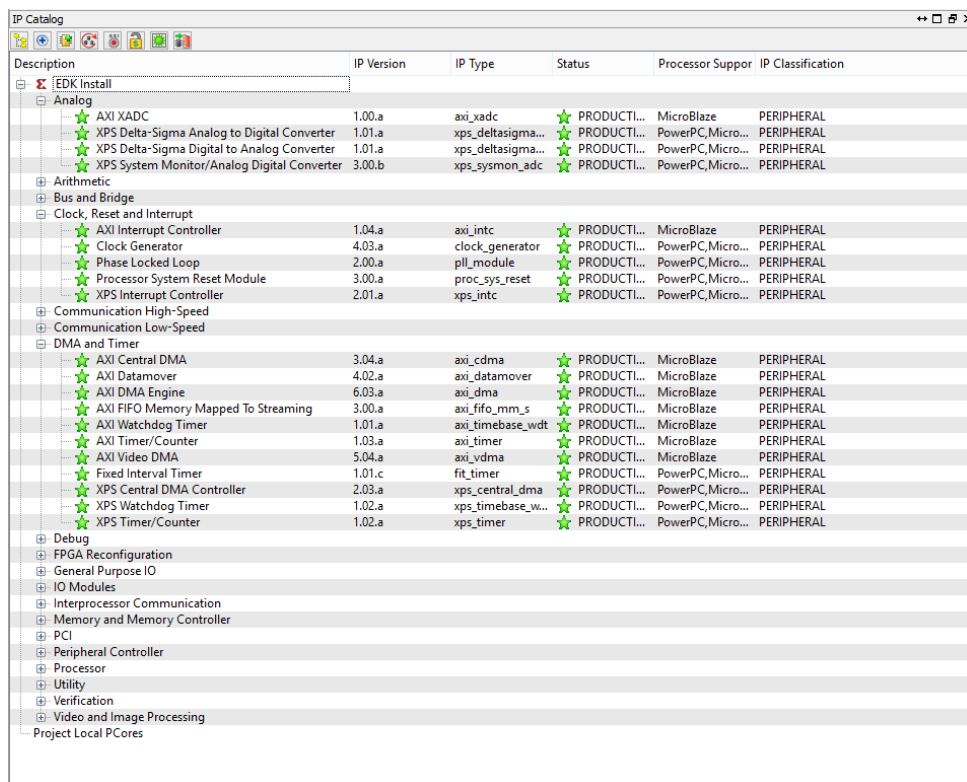
2.3. MicroBlaze

MicroBlaze és el nom que rep la implementació de microcontrolador soft-core del fabricant Xilinx. Aquest és un microcontrolador de tipus RISC (de l'anglès Reduced Instruction Set Computing), amb un nombre reduït d'instruccions de codi màquina. Aquests tipus de microcontroladors estan dissenyats per ser el més ràpid possibles amb la filosofia de tenir instruccions més senzilles d'executar sobre hardware suficientment potent com per executar-les fent servir el mínim nombre de cicles de rellotge. El MicroBlaze fa servir un bus d'interconnexió AXI (Advanced eXtensible Interface) entre els seus perifèrics i memòria, igual que els microcontroladors ARM més moderns. Això fa que sigui relativament fàcil de programar per aquest microcontrolador, utilitzant un llenguatge d'alt nivell¹ com podria ser C.

2.3.1. Generació de Hardware

La generació de hardware per una FPGA del fabricant Xilinx es fa mitjançant una eina del mateix fabricant (Xilinx Platform Studio) que permet incorporar IP cores a un disseny de hardware com si es tractés d'una llista seleccionable. Aquesta eina consta de tot el necessari per triar els IP cores que l'usuari necessita i fer tot el disseny, generació d'arxius i compilació perquè el següent pas sigui programar pel microcontrolador MicroBlaze en llenguatge C/C++.

Un exemple dels IP cores que es poden seleccionar en aquest programari es pot trobar a la Figura 6. Com es pot veure, existeixen gran varietat de IP cores: controladors d'interrupcions, controladors d'accés directe a memòria (DMA) i IP cores per depurar codi sobre MicroBlaze, entre d'altres.



Description	IP Version	IP Type	Status	Processor Support	IP Classification
Analog					
AXI XADC	1.00.a	axi_xadc	PRODUCT...	MicroBlaze	PERIPHERAL
XPS Delta-Sigma Analog to Digital Converter	1.01.a	xps_deltasigma...	PRODUCT...	PowerPC, Micro...	PERIPHERAL
XPS Delta-Sigma Digital to Analog Converter	1.01.a	xps_deltasigma...	PRODUCT...	PowerPC, Micro...	PERIPHERAL
XPS System Monitor/Analog Digital Converter	3.00.b	xps_sysmon_adc	PRODUCT...	PowerPC, Micro...	PERIPHERAL
Arithmetic					
Bus and Bridge					
Clock, Reset and Interrupt					
AXI Interrupt Controller	1.04.a	axi_intc	PRODUCT...	MicroBlaze	PERIPHERAL
Clock Generator	4.03.a	clock_generator	PRODUCT...	PowerPC, Micro...	PERIPHERAL
Phase Locked Loop	2.00.a	pll_module	PRODUCT...	PowerPC, Micro...	PERIPHERAL
Processor System Reset Module	3.00.a	proc_sys_reset	PRODUCT...	PowerPC, Micro...	PERIPHERAL
XPS Interrupt Controller	2.01.a	xps_intc	PRODUCT...	PowerPC, Micro...	PERIPHERAL
Communication High-Speed					
Communication Low-Speed					
DMA and Timer					
AXI Central DMA	3.04.a	axi_cdma	PRODUCT...	MicroBlaze	PERIPHERAL
AXI Datamover	4.02.a	axi_datamover	PRODUCT...	MicroBlaze	PERIPHERAL
AXI DMA Engine	6.03.a	axi_dma	PRODUCT...	MicroBlaze	PERIPHERAL
AXI FIFO Memory Mapped To Streaming	3.00.a	axi_fifo_mm_s	PRODUCT...	MicroBlaze	PERIPHERAL
AXI Watchdog Timer	1.01.a	axi_timebase_wdt	PRODUCT...	MicroBlaze	PERIPHERAL
AXI Timer/Counter	1.03.a	axi_timer	PRODUCT...	MicroBlaze	PERIPHERAL
AXI Video DMA	5.04.a	axi_vdma	PRODUCT...	MicroBlaze	PERIPHERAL
Fixed Interval Timer	1.01.c	fit_timer	PRODUCT...	PowerPC, Micro...	PERIPHERAL
XPS Central DMA Controller	2.03.a	xps_central_dma	PRODUCT...	PowerPC, Micro...	PERIPHERAL
XPS Watchdog Timer	1.02.a	xps_timebase_w...	PRODUCT...	PowerPC, Micro...	PERIPHERAL
XPS Timer/Counter	1.02.a	xps_timer	PRODUCT...	PowerPC, Micro...	PERIPHERAL
Debug					
FPGA Reconfiguration					
General Purpose IO					
IO Modules					
Interprocessor Communication					
Memory and Memory Controller					
PCI					
Peripheral Controller					
Processor					
Utility					
Verification					
Video and Image Processing					
Project Local PCores					

Figura 6. Extracte d'una llista de IP cores disponibles per l'usuari

¹ Alt nivell en comparació amb el llenguatge d'assemblador.

Un cop es tenen seleccionats els IP cores que necessita l'usuari pel seu disseny aquests es mostren en una altra part del programari on es pot veure com estaran organitzats segons els busos d'interconnexió entre els mateixos (busos hardware). En aquesta altra vista també es poden configurar paràmetres dels mòduls individualment, triar en quines adreces de memòria estaran situats, etcètera. Aquesta part del programari es pot veure a la Figura 7.

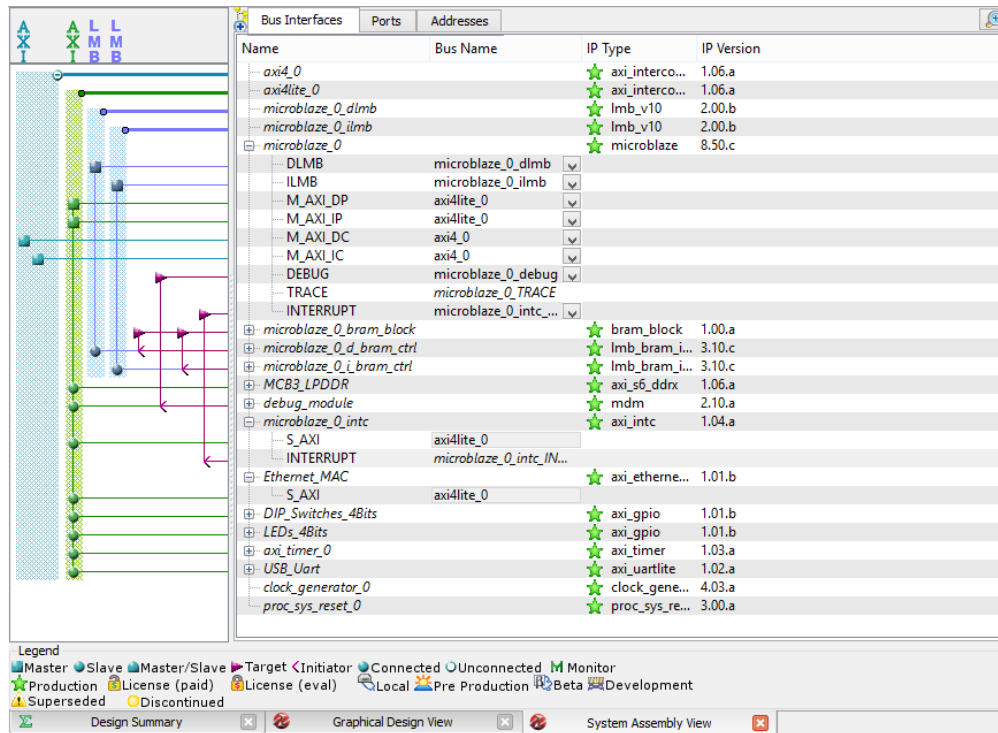


Figura 7. Vista de configuració dels mòduls IP core triats pel projecte en curs

Una vista útil més del programari és la d'assignació de adreces de memòria del disseny. En aquest cas, com la placa utilitzada té una memòria LPDDR de 64 MB i MicroBlaze no té cap problema per executar codi des de memòria RAM, s'utilitza el IP core per controlar aquesta memòria per tenir un espai pràcticament il·limitat pel codi del programa (tenint en compte que l'aplicació en qüestió és *petita*).

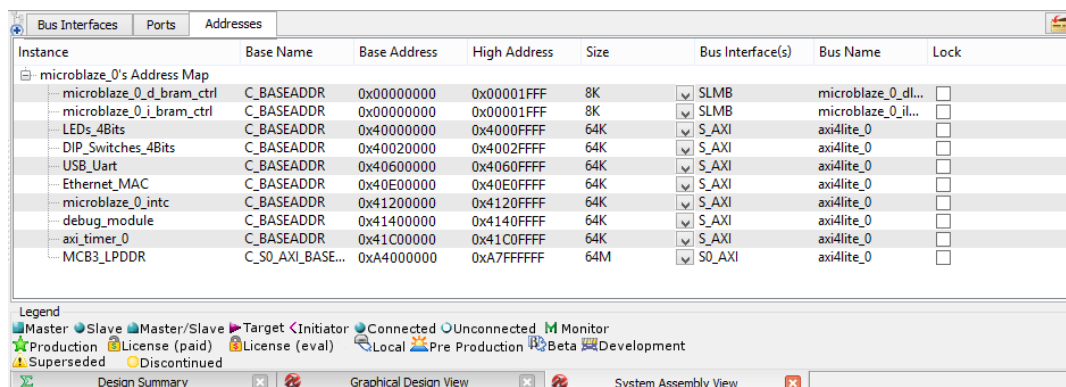


Figura 8. Extracte d'una llista de IP cores disponibles per l'usuari

Un cop tot el disseny ha estat configurat correctament, per poder utilitzar-lo per carregar-lo a una FPGA l'usuari ha de generar un arxiu conegut com *bitstream*. Aquest és l'arxiu de configuració de la lògica de la FPGA. Un cop es té aquest arxiu es pot carregar a la FPGA i el hardware es configurarà tal com l'usuari l'ha dissenyat.

Aleshores, com es té un microcontrolador al sistema, aquest mateix s'ha de programar per executar el codi de l'usuari. Abans de poder fer aquesta programació, però, s'ha d'exportar el projecte necessari per fer servir aquest hardware generat des de l'entorn de desenvolupament de software.

2.3.2. Programació de Software

Un cop es té generat el programa que farà que la FPGA quan sigui programada es comporti a nivell electrònic segons s'ha dissenyat, cal desenvolupar el codi pel microcontrolador que s'ha decidit incloure al disseny. Això es fa des d'un altre paquet del programari que ofereix el fabricant (Xilinx Software Development Kit). Dins d'aquest programari es requereix importar el projecte que s'ha exportat del programari descrit prèviament. Això porta a l'usuari a tenir un projecte que defineix una plataforma hardware. Un cop es té aquest projecte dins l'espai de treball es requereix crear un altre tipus de projecte que dona accés a tota la propietat intel·lectual necessària (llibries) per controlar els perifèrics dels IP cores des del software que executarà el MicroBlaze. Aquest segon projecte s'anomena Board Support Package (BSP). Un cop creat el BSP, l'usuari pot crear tants projectes d'aplicació com necessiti, on programarà el seu software com per qualsevol altre microcontrolador, en llenguatge C/C++. Es pot veure una captura de pantalla d'aquest programari a la Figura 9. En aquesta figura, a la pestanya Project Explorer de l'esquerra es poden veure els tres projectes descrits anteriorment: *app*, *standalone_bsp_0* i *xps_hw_platform*.

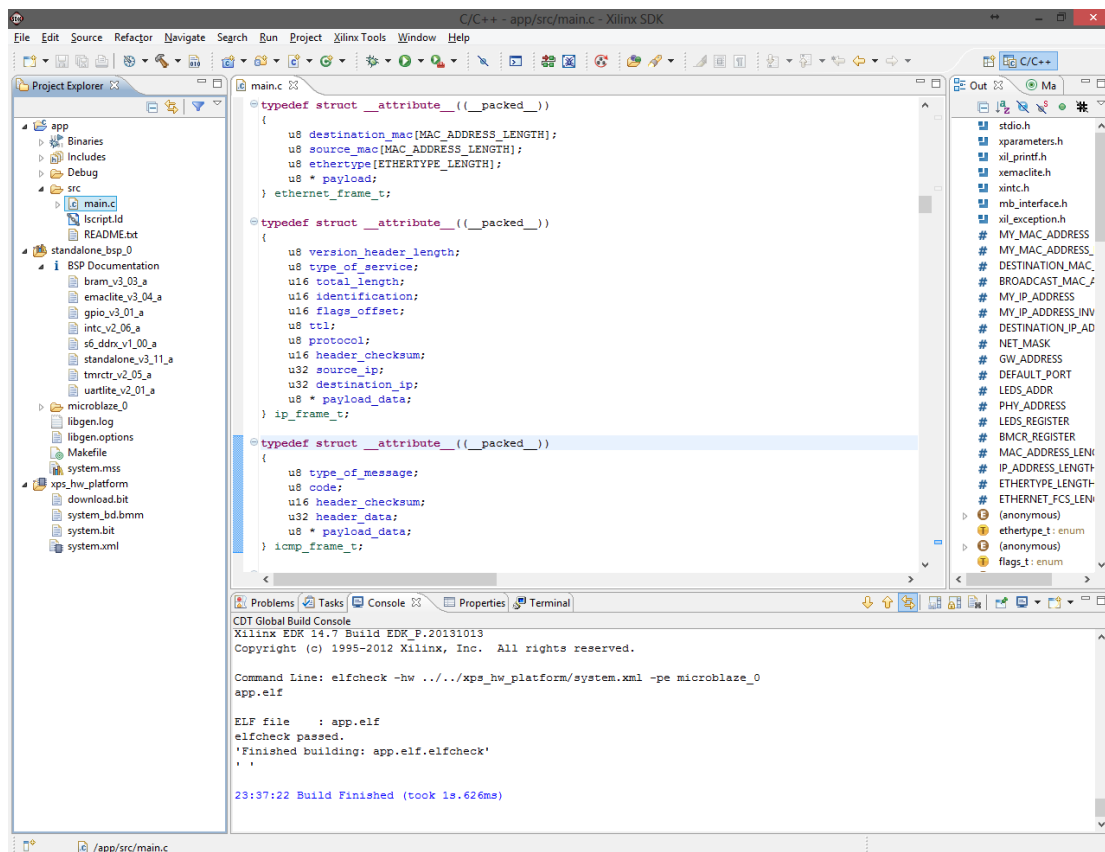


Figura 9. Captura de pantalla del programari Software Development Kit

3. ETHERNET

3.1. Pila de protocol per capes

Com s'ha dit prèviament, Ethernet en sí és un estàndard que només s'aplica sobre les capes física i d'accés al medi en una xarxa d'equips interconnectats. Això vol dir que Ethernet no tracta les dades *útils* de cap manera. Generalment, són els protocols de nivells superiors els que tracten amb les dades. Tots aquests protocols, dividits per les anomenades capes, estan englobats dins de l'estàndard conegut com model OSI (de l'anglès Open Systems Interconnection) de l'Organització Internacional per a l'Estandardització (ISO). Aquest model defineix les capes de protocol que es necessita per connectar-se a la xarxa de xarxes, Internet. Una representació del model OSI es pot trobar a la Figura 10.

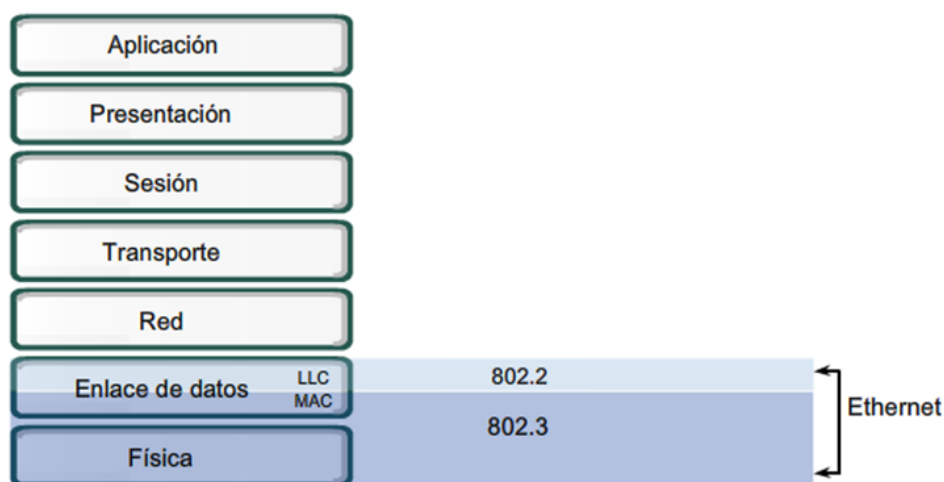


Figura 10. Representació del model OSI

Ethernet és l'encarregat de lligar els sistemes físicament i a nivell de paquets de dades, tal com es descriu a les normatives IEEE 802.3 i 802.2. A la capa física s'hi troba el tipus de connexió que s'ha d'utilitzar per tenir connectivitat Ethernet. Generalment la connectivitat física d'Ethernet són cables amb parells diferencials trenats. Segons la taxa de bits que es vol fer servir, hi ha diferents estàndards dintre de la pròpia connectivitat Ethernet. Generalment els estàndards són de 10/100/1000 Mbit/s. El nombre de parells trenats també depèn de si es vol comunicació full-duplex o n'hi ha suficient amb half-duplex. Aquests estàndards requereixen de diferents tipus de cablejat. Per exemple, a la Figura 11 es pot veure una comparativa entre cables de diferents categories, que serveixen per les diferents velocitats. Les diferències entre uns cables i uns altres generalment són: la quantitat de parells diferencials disponibles, la millora en aïllament entre aquests i la densitat del trenat dels mateixos. Per exemple, per funcionar a 10 Mbit/s no fan falta més que cables de categoria 3, amb 2 parells trenats, mentre que per funcionar a 100 Mbit/s es necessiten cables de categoria 5 com a mínim. També existeixen cables de categories superiors, que permeten velocitats de transmissió superiors als 1000 Mbit/s, arribant en alguns casos fins a 10 Gbit/s.

Category Cable Wiring

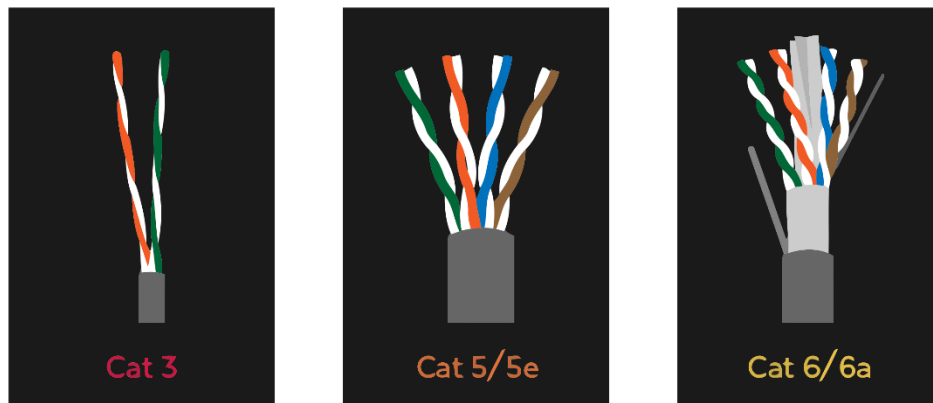


Figura 11. Diferents categories de cablejat per Ethernet que es poden trobar

En aquesta figura es pot veure que els parells dels cables de categoria 3 estan trenats molt poc densament i no tenen cap tipus d'aïllament electromagnètic, ni entre ells ni respecte a l'ambient. A l'altre extrem es troben els cables de categoria 6a, que estan aïllats front a interferències electromagnètiques mitjançant apantallament del cable amb malla de coure, i també entre ells fent servir una pel·lícula metal·litzada d'alumini-mylar.

En una xarxa Ethernet tots els equips connectats tenen la seva pròpia adreça física, també coneguda com adreça d'accés al medi (MAC). Aquesta adreça ha de ser única segons el tipus de dispositiu. Es tracta d'una adreça de 6 bytes on els 3 primers identifiquen el fabricant del dispositiu connectat a la xarxa, i els altres 3 identifiquen el model del dispositiu fabricat per aquell fabricant. Normalment s'expressa en notació hexadecimal (base 16), amb el valor dels bytes separats per dos punts. Per exemple, en el cas de Xilinx, l'adreça de fabricant que li correspon és: 00:35:0a.

Generalment, les xarxes Ethernet segueixen una topologia en estrella, on tots els equips estan connectats a un enrutador que dirigeix els paquets. Però això no és cap requisit. Es pot tenir, per exemple, una connexió punt a punt entre dos equips, o una connexió en anell si hi ha varis equips a connectar. A la Figura 12 es mostren algunes topologies utilitzades en les xarxes.

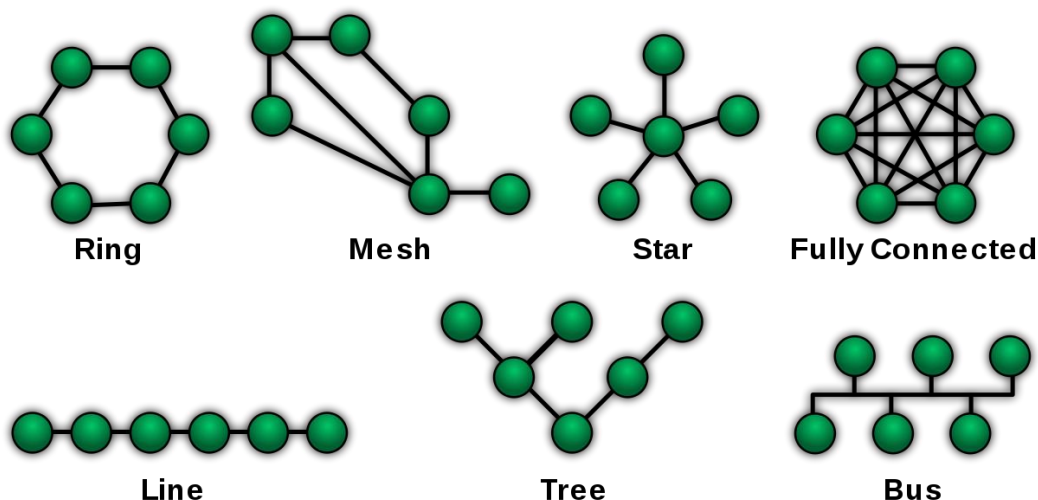


Figura 12. Representació de diverses topologies de xarxes possibles

3.2. Funcionament d'Ethernet a la placa utilitzada

El funcionament d'Ethernet a la placa utilitzada és similar al de tots els dispositius que compleixin l'estàndard. Es tracta bàsicament d'un circuit integrat que interpreta els senyals que arriben a través dels parells trenats del cable. A més d'aquest xip, es necessita una capa de hardware/software que gestiona l'accés al medi (MAC, de l'anglès Medium Access Control). En aquest cas, la capa MAC està integrada a la lògica de la FPGA a través d'un IP core de Xilinx, EmaLite. Aquesta capa és qui es comunica amb el circuit integrat. Una descripció gràfica es pot veure a la Figura 16.

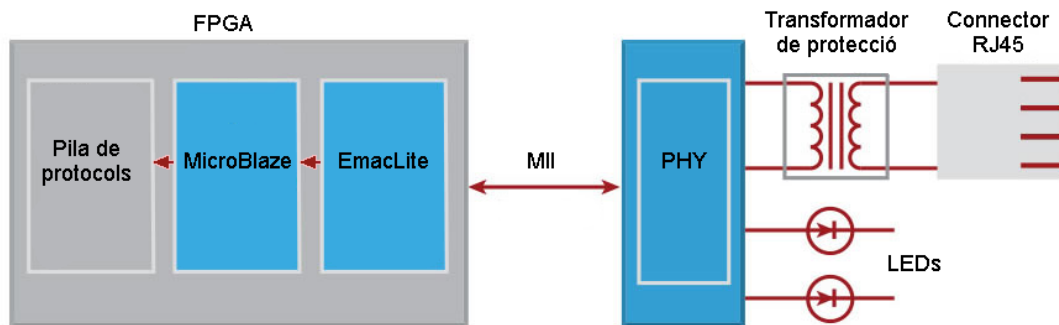


Figura 13. Diagrama que representa el funcionament d'Ethernet a la placa utilitzada

El circuit integrat que fa de PHY (de l'anglès PHYsical layer, en referència a la capa física del model OSI) en la placa utilitzada és el DP83848J de Texas Instruments. Segons el datasheet d'aquest xip, és capaç de comunicar-se amb la xarxa a un màxim de 100 Mbit/s, i gestiona automàticament la negociació de velocitat de transmissió amb la resta de dispositius de la xarxa, segons la normativa IEEE 802.3. Aquest xip es comunica amb la FPGA a través d'un bus sèrie anomenat MII, de l'anglès Media Independent Interface. Aquest és un bus sèrie de 4 bits en paral·lel per transmissió i 4 bits en paral·lel per recepció de dades. Això fa que no necessiti freqüències massa altes per aconseguir aquestes taxes de bits. Utilitzant rellotges de 25 MHz n'hi ha suficient per aconseguir 100 Mbit/s. A més d'aquests 8 bits també es fan servir una sèrie de senyals de gestió d'aquesta comunicació, per gestionar-la el més eficientment i ràpida possible. Al nivell del programa del MicroBlaze, aquesta comunicació la gestiona una llibreria proporcionada per Xilinx. Així doncs, l'usuari és responsable de posar en marxa els perifèrics i configurar-los correctament. Aquest funcionament es descriurà amb més detall al capítol 4.

3.3. Protocols utilitzats

En aquest treball es fan servir varis protocols que són transportats per la xarxa Ethernet. Com s'ha dit anteriorment, Ethernet no treballa amb les dades i per tant només és un *mitjà de transport* de les trames necessàries per intercanviar dades entre els equips connectats a la xarxa. Els protocols que es fan servir estan encapsulats dins d'aquestes trames. Cal notar que aquests protocols no són exclusius d'Ethernet, i que són els que també es fan servir a través de la xarxa de xarxes, Internet. Alguns dels protocols més coneguts i utilitzats són:

- IP (Internet Protocol), engloba tots els protocols necessaris per tal que la xarxa de xarxes pugui funcionar.
- ARP (Address Resolution Protocol), que serveix per descobrir les adreces físiques (adreces MAC) que té un dispositiu a la xarxa.
- DHCP (Dynamic Host Configuration Protocol), que serveix per tal de que els clients en una xarxa puguin ser configurats automàticament per l'amfitrió per poder connectar-se a Internet i ser identificats pels altres equips d'aquesta xarxa.
- TCP (Transfer Control Protocol), és un dels protocols d'intercanvi de dades a Internet més utilitzats. Conté funcions de reenviament i detecció/control d'errors en les comunicacions. És un protocol orientat a connexió, és a dir, tots dos equips s'han de posar d'acord per comunicar-se, i tots dos han de confirmar les dades rebudes.
- UDP (User Datagram Protocol), un altre dels protocols més utilitzats a Internet. No és orientat a connexió i permet enviar i rebre dades a més altes velocitats que TCP, degut a què les dades no són crítiques. Aquest protocol és més eficient que TCP perquè *desaprofita* menys bytes transmesos degut a què no es controlen errors ni recepció de dades.
- ICMP (Internet Control Message Protocol), és un dels protocols més bàsics que permet regular el tràfic a la xarxa. És el protocol fonamental per poder, per exemple, comunicar missatges d'error, notificar que un servei determinat no està disponible, o provar si un equip es troba a la xarxa o no.

A continuació es descriuen amb cert detall els protocols que s'utilitzaran més endavant al treball, en la secció 4.

3.3.1. Trames Ethernet

Per tal de poder transmetre dades a través del medi físic es requereix que les trames Ethernet tinguin un format específic. En aquest cas, és un format simple on s'ha d'enviar:

- Un preàmbul que serveix per sincronitzar els equips que s'estiguin comunicant en aquell moment.
- L'adreça física (MAC) de destí de les dades encapsulades en aquella trama.
- L'adreça física (MAC) d'origen de les dades encapsulades en aquella trama.
- El tipus de dades (EtherType) encapsulades en aquella trama. Els valors són estandarditzats. Per exemple el valor 0x0800 indica que les dades encapsulades corresponen a IPv4, mentres que 0x0806 indica que les dades són ARP.
- Les dades encapsulades. Si la llargària total del paquet encapsulat no arriba als 46 bytes s'ha de omplir amb valors nuls.
- Una seqüència de control de trama (FCS, de l'anglès Frame Check Sequence), consistent del resultat d'executar un algorisme de comprovació de redundància

cíclic (CRC) de 32 bits (4 bytes) sobre tota la trama. Generalment aquest algoritme és transparent a l'usuari i s'executa a nivell de hardware a la capa MAC del dispositiu connectat a la xarxa.

A la Figura 13 es pot veure una representació gràfica d'aquesta trama.

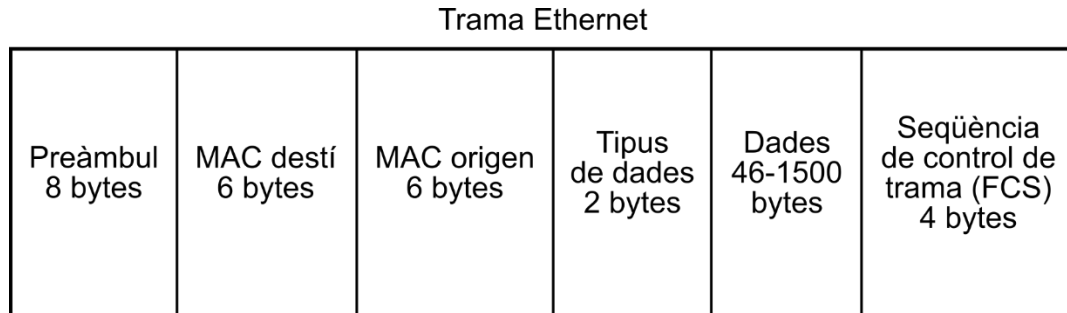


Figura 14. Diagrama que representa la estructura d'una trama Ethernet

A continuació es descriuen els protocols que van encapsulats dins de la trama Ethernet, i que permeten intercanviar dades *útils* entre els equips connectats a la xarxa.

3.3.2. Paquets IPv4

El protocol bàsic que s'utilitza per l'intercanvi de dades tant dins d'una xarxa local com a la xarxa de xarxes és IP versió 4. 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 física, que permet identificar-los independentment del dispositiu, i permet organització lògica dels equips, mitjançant formació de xarxes i subxarxes. L'adreça IP consta de 4 bytes de llargària. En aquest protocol els paquets de dades se solen separar entre una capçalera, que dóna informació sobre l'origen i destí del paquet entre d'altres, i les dades encapsulades en aquest paquet. La capçalera dels paquets IPv4 ha de contenir, com a mínim:

- La versió de protocol IP que s'està utilitzant (4 o 6), 4 bits.
- La llargària de la capçalera IP, comptada en paraules de 4 bytes (32 bits), 4 bits. Això dóna un màxim de 15, per tant $15 \cdot 32 \text{ bits} = 480 \text{ bits} = 60 \text{ bytes}$. És a dir, una capçalera de IPv4 mai hauria de ser més llarga de 60 octets.
- Un camp anomenat Serveis Diferenciats de 6 bits. Aquest camp permet distingir entre tipus de trames IPv4, per tractar-les amb major o menor prioritat, per exemple.
- Un camp anomenat Notificació de Congestió Explícita, de 2 bits. Aquest camp serveix per, si hi ha congestió a la xarxa, es pugui notificar als equips de destí i origen, sense pèrdua de paquets. Generalment aquests protocols no s'utilitzen amb freqüència.
- La llargària total del paquet IP, incloent les dades, comptant bytes. Aquest és un camp de 16 bits, i per tant permetria que els paquets tinguin una llargària màxima de fins a 65535 bytes.
- Un camp anomenat Identificació, que permet identificar els paquets, de 16 bits. Com que aquesta identificació hauria de ser única pel que duri la connexió i transmissió entre els dos equips de la xarxa, això limita altament les velocitats de transmissió, així que generalment no s'utilitza per res.
- El camp anomenat Flags, de 3 bits. Aquest camp té un bit que indica si el paquet IPv4 ha hagut de ser fragmentat en vàries transmissions diferents (MF, de l'anglès More Fragments), si les dades no cabessin en un sol paquet, i per tant han de venir més fragments. També té un bit (DF, de l'anglès Don't

Fragment) segons el qual l'originador del paquet pot demanar que no se separin les dades durant la transmissió, saltant pels diferents punts de la xarxa. El tercer bit està reservat, segons la especificació.

- El camp que indica, si haguessin d'haver més fragments, quin nombre de fragment s'està tractant en el paquet actual. Aquest camp té una llargària de 13 bits, però el seu número es mesura en unitats de 8 bytes. El primer paquet de la transmissió sempre comença tenint un 0.
- El *temps de vida* que li queda al paquet, 8 bits. Aquest temps de vida es mesura restant d'un nombre màxim. Es resta un per cada salt que fa aquest paquet a la xarxa, entenent com salt passar d'un equip a un altre. Això impedeix que un paquet quedi donant voltes per la xarxa de forma indefinida.
- Un camp que defineix el protocol de les dades encapsulades en aquest paquet IPv4, de 8 bits. Alguns exemples són: 0x01 pel protocol ICMP, 0x11 per UDP, 0x06 per TCP.
- El camp que comprova la suma de verificació de la capçalera del paquet en qüestió, de 16 bits. Aquest valor s'obté calculant el complement a u, de la suma de tota la capçalera en complement a u, assumint com si la capçalera estigués formada per paraules de 16 bits.
- L'adreça IP d'origen del paquet, de 32 bits (4 bytes).
- L'adreça IP de destí del paquet, de 32 bits (4 bytes).
- Per últim, el camp de les dades encapsulades en aquest paquet, de llargària variable.

A la Figura 14 es pot veure una representació gràfica d'aquest paquet.

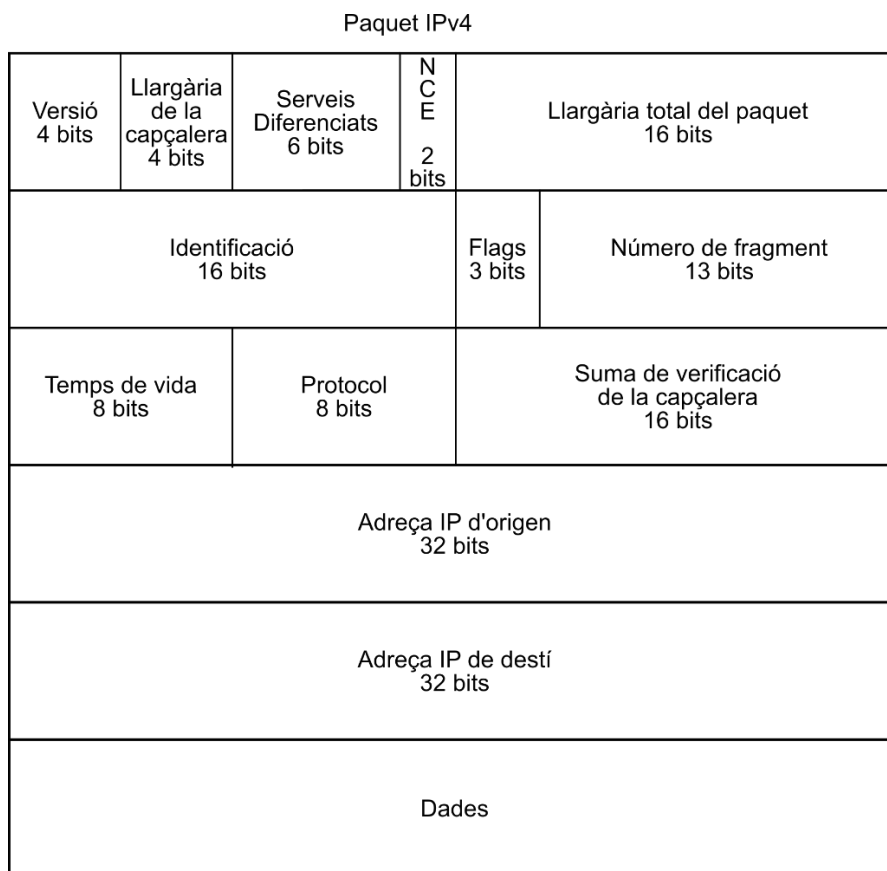


Figura 15. Diagrama que representa la estructura d'un paquet IPv4

A continuació es descriuen els protocols que es faran servir més endavant, encapsulats dins del paquet IPv4.

3.3.3. Paquet ARP

El protocol ARP serveix per resoldre les adreces físiques 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), fent servir com a adreça MAC de destí l'adreça FF:FF:FF:FF:FF:FF. Els paquets ARP es componen de les següents dades:

- Medi a què s'està accedint, 16 bits. Per una xarxa Ethernet, aquest valor ha de ser 0x0001.
- Tipus de protocol que s'utilitza a la xarxa a què s'accedeix, 16 bits. Per IPv4, el valor ha de ser 0x0800. Normalment s'utilitzen els mateixos valors que el EtherType de la trama Ethernet.
- Llargària de les adreces físiques (MAC), comptant bytes, 8 bits. Per Ethernet, aquest camp ha de ser 6.
- Llargària de les adreces lògiques (IP), comptant bytes, 8 bits. Per IPv4, aquest camp ha de ser 4.
- Codi d'operació, 16 bits. Codi que defineix què ha de fer el receptor amb aquest paquet. Pot ser una petició (1) o una resposta (2).
- Adreça física de qui envia el paquet, llargària 6 bytes.
- Adreça lògica de qui envia el paquet, llargària 4 bytes.
- Adreça física del destinatari del paquet, llargària 6 bytes. Quan no se sap i es vol descobrir l'adreça física del dispositiu ubicat a una determinada adreça lògica, aquest camp es plena amb valors nuls.
- Adreça lògica del destinatari del paquet, llargària 4 bytes.

A la Figura 14 es pot veure una representació gràfica d'aquest paquet.

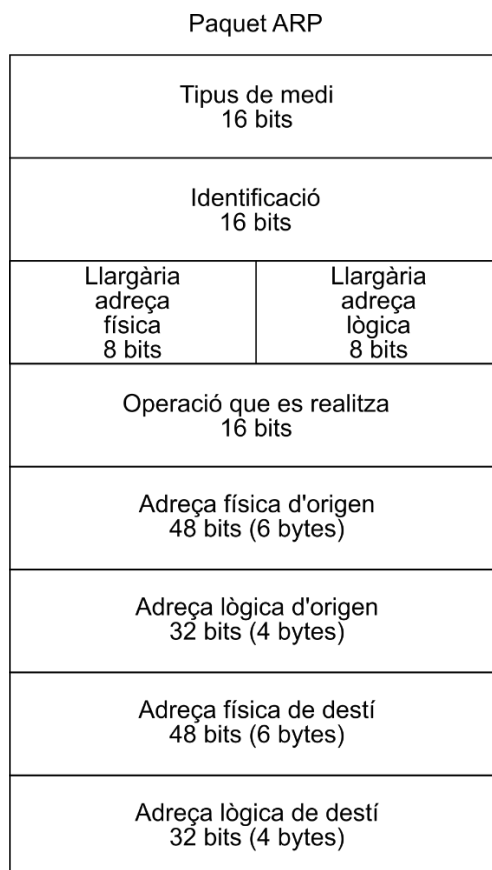


Figura 16. Diagrama que representa la estructura d'un paquet ARP

3.3.4. Paquet ICMP

El protocol ICMP 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. Aquesta funció serà la que s'utilitzarà extensament als capítols 4 i 5 per fer les proves comparant entre les diferents implementacions de piles de protocol. El paquet ICMP està compostat per les següents dades:

- Tipus de missatge que es fa servir, 8 bits. Alguns d'aquests tipus són, per exemple: 0 per una resposta a un paquet eco, 3 quan el destinatari d'algun paquet és inabastable, 8 per una petició d'eco, i 11 per informar que un paquet ha arribat a 0 del seu temps de vida.
- Codi que acompanya el tipus de missatge, 8 bits. Aquest codi dona més detalls sobre el tipus de missatge del que s'està informant.
- Suma de verificació del paquet ARP, 16 bits. És el mateix algoritme que el que comprova la capçalera IPv4, però en aquest cas s'aplica a tot el paquet ARP.
- Identificador i nombre de seqüència, 32 bits, generalment utilitzats per associar una petició d'eco amb una resposta d'eco.
- Altres dades. Aquí s'hi podria incloure una data i hora d'enviament del missatge (si es tractés d'una petició d'eco), per exemple. El receptor de la petició d'eco retorna aquestes dades sense modificació. Això fa que quan l'originador de la petició d'eco inicial rebi les dades sàpiga quant de temps ha passat, degut a què la seva referència de data i hora no ha canviat, així que les pot comparar directament.

Paquet ICMP

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

Figura 17. Diagrama que representa la estructura d'un paquet ARP

4. IMPLEMENTACIÓ PRÀCTICA

4.1. LwIP

4.2. Pila programada per l'autor

5. RESULTATS EXPERIMENTALS

5.1. Mètode d'estudi

5.2. Resultats estadístics

5.3. Comparació entre les piles

6. CONCLUSIONS I TREBALL FUTUR

6.1. Conclusió

6.2. Treball futur

7. BIBLIOGRAFIA I ANNEXES

AGRAÏMENTS