****

**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**

Amb una extensió màxima de 50 línies, i amb una llista de màxim 10 paraules clau, el resum és un text informatiu que permet decidir sobre la utilitat de llegir el document complet; ha de definir l’objectiu, els mètodes, els resultats i les conclusions presentats en el cos del document, en aquest ordre o destacant inicialment els resultats i les conclusions; ha de ser un text complet perquè sigui intel·ligible sense necessitat de referir-se a la memòria; ha de contenir la informació bàsica i el caràcter del document original. Com en tots els documents cal vetllar per la correcció d’estil, cal també emprar una nomenclatura normalitzada, i definir els termes no familiars les abreviacions i els símbols, quan apareguin per primera vegada en el resum. És la pàgina número 1 del document.

**Paraules clau (màxim 10):**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**ABSTRACT**

**Keywords (10 maximum):**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**SUMARI**

1. Introducció8
   1. Objectius8
2. FPGA9
   1. Estructura9
   2. Nuclis de propietat intel·lectual11
   3. MicroBlaze12
      1. Generació de Hardware12
      2. Programació de Software14
3. Ethernet15
   1. Capes de la pila de protocol15
   2. Funcionament a la placa utilitzada16
4. Implementació pràctica18
   1. LwIP5
   2. Pila programada per l’autor5
5. Resultats experimentals9
   1. Mètode d’estudi5
   2. Resultats estadístics5
   3. Comparació entre les piles5
   4. Conclusions5
6. conclusions i treball futur9
   1. Conclusió5
   2. Treball futur5
7. Bibliografia i Annexes9

**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

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. Objectius

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

1. FPGA
   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 . 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 . 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 . 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 . Representació simplificada d’un dels blocs d’entrada/sortida de la FPGA

* 1. 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 . Foto de la placa utilitzada en aquest treball

* 1. 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 nivell[[1]](#footnote-1) com podria ser C.

* + 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.



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

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.

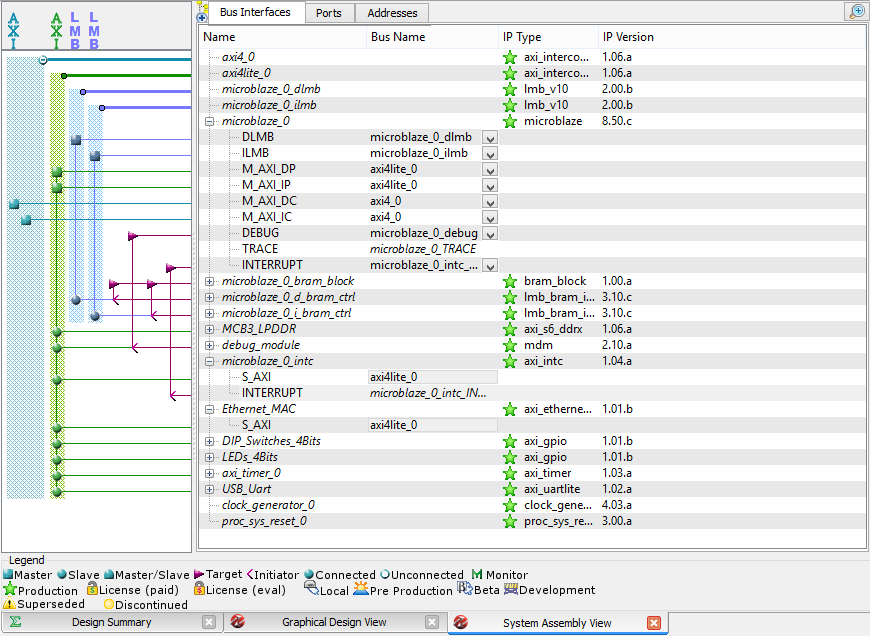


Figura 7. Vista de configuració dels mòduls IP core triats pel projecte en curs

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*).

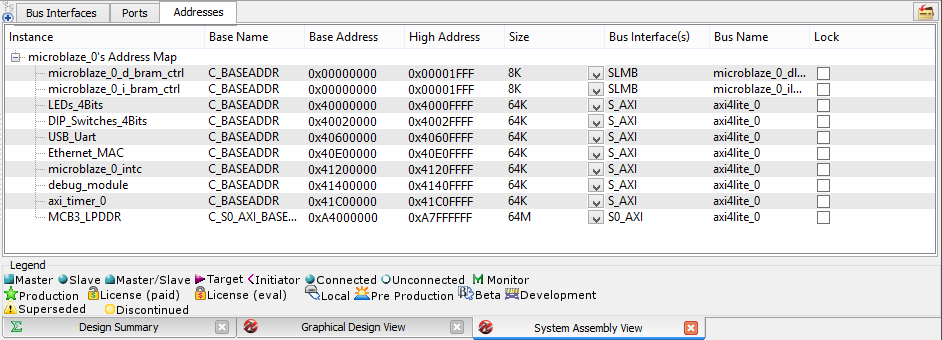


Figura 8. Extracte d’una llista de IP cores disponibles per l’usuari

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.

* + 1. 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 dóna accés a tota la propietat intel·lectual necessària (llibreries) 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

1. Ethernet
   1. Capes de la pila de protocol

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 . 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.



Figura . 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.

* 1. Funcionament 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, EmacLite. Aquesta capa és qui es comunica amb el circuit integrat. Una descripció gràfica es pot veure a la Figura 12.



Figura 12. 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.

1. Implementació PRàctica

CONCLUSIONS

Les conclusions han de ser un reflex clar i ordenat de les deduccions fetes com a conseqüència del treball descrit al llarg del nucli del cos de la memòria. Es poden incloure dades quantitatives però no s’haurien de donar detalls de cap argument o resultat.

Les recomanacions són manifestacions concises d’alguna acció futura que sembli necessària, com a resultat directe de les conclusions o d’alguna experiència feta en el curs del treball objecte del projecte. No són necessàries, tret que estiguin completament justificades pel treball descrit. Aquest apartat no format part del cos del document i no necessita portar numeració de capítol

AGRAÏMENTS

Es poden incloure agraïments relatius a ajuts en la realització del treball i en la preparació del dcoument. No és habitual agrair les contribucions com ara un control de rutina, un petit ajut o uns agraïments de caràcter general. El reconeixement d’altres treballs emprats ha de fer-se en forma de referència. Els agraïments que fan referència a un text citat a i l’ús de taules i il·lustracions poden requerir reconeixement de drets d’autor.

BIBLIOGRAFIA

Únicament han de figurar en aquest apartat aquelles referències bibliogràfiques que hagin estat citades al llarg del TFG/TFM. Les entrades o els elements de la llista de referències han de donar-se segons l’esquema general Autor/Títol/Dades de la publicació

Per més informació us recomanem visiteu la pàgina web de Publica (que fa referència a l’elaboració de referències bibliògrafiques: http://publica.upc.edu/ca/estil/iso690

Exemples:

CARDONA, S. Teoria de màquines. Barcelona, Edicions UPC, 2000, p. 99-105

GUTOWSKI, T,G., DYM, C.L. Propagation of ground vibration: a review. Journal of Sound and Vibration. Vol. 49(2), 1976, p. 179-193.

ANNEX

1. Alt nivell en comparació amb el llenguatge d’assemblador. [↑](#footnote-ref-1)