



UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH



Unitats funcionals aproximades per a processadors de baix consum

Autor: Martí Caro Roca

Director: Ramon Canal Corretger
Departament: Arquitectura de computadors

Titulació: Grau en Enginyeria Informàtica
Especialitat: Enginyeria de Computadors

Centre: Facultat d'informàtica de Barcelona (FIB)
Universitat: Universitat Politècnica de Catalunya (UPC) - BarcelonaTech

Quadrimestre: Tardor 2019-2020

Resum

Actualment els multiplicadors són una de les unitats funcionals que requereixen més consum d'energia degut al gran nombre de portes lògiques que contenen. Aprofitant el fet que diverses aplicacions poden tolerar un cert error en el resultat del càlcul de les multiplicacions, principalment aplicacions relacionades amb els sentits de l'ésser humà ja que els nostres sentits no són perfectes, podem implementar multiplicadors aproximats que realitzen el càlcul de la multiplicació de manera aproximada i permeten reduir el consum d'energia, mantenint un error relatiu petit en el resultat de l'operació.

En aquest projecte s'estudien diverses implementacions de multiplicadors aproximats de 8, 16 i 32 bits i també s'ha estudiat l'efecte de combinar multiplicadors aproximats en una mateixa implementació.

Resumen

Actualmente los multiplicadores son una de las unidades funcionales que requieren más consumo de energía debido al gran número de puertas lógicas que contienen. Aprovechando el hecho que diversas aplicaciones pueden tolerar un cierto error en el resultado de los cálculos de las multiplicaciones, principalmente aplicaciones relacionadas con los sentidos del ser humano puesto que nuestros sentidos no son perfectos, podemos implementar multiplicadores aproximados que realizan el cálculo de las multiplicaciones de forma aproximada y permiten reducir el consumo de energía, manteniendo un error relativo pequeño en el resultado de la operación.

En este proyecto se estudian diversas implementaciones de multiplicadores aproximados de 8, 16 y 32 bits y también se estudia el efecto de combinar diferentes implementaciones de multiplicadores aproximados en una misma implementación.

Abstract

Currently multipliers are one of the functional units that consume more energy due to the large number of logic gates used. Taking advantage of the fact that some applications tolerate a small error on the result of the multiplications, mainly applications related to the human senses since our senses are not perfect, we can implement approximate multipliers that calculate the result of the multiplications approximately and, hence, reduce the energy needed while maintaining a small relative error in the result of the operation.

In this project, we study different approximate multiplier implementations of 8, 16 and 32 bits and we also study the effect of combining different implementations of approximate multipliers in one implementation.

Índex

1 Contextualització i abast del projecte	8
1.1 Context i formulació del problema	8
1.2 Actors implicats	9
1.2.1 Desenvolupador	9
1.2.2 Director	9
1.2.3 Beneficiaris	9
1.3 Justificació	10
1.3.1 Estudis similars	10
1.4 Abast del projecte	11
1.4.1 Objectius	11
1.4.2 Possibles obstacles i riscos	11
1.5 Metodologia i rigor	12
1.5.1 Eines de treball	12
1.5.2 Eines per monitoritzar l'evolució del projecte	13
1.5.3 Mètodes de validació	13
2 Estudi dels multiplicadors aproximats	13
2.1 Procés d'implementació i avaluació dels multiplicadors	14
2.2 Wallace tree	15
2.3 AM1	16
2.4 AM2	17
2.5 AM3	19
2.6 AM4	19
2.7 Mètriques de l'estudi	19
3 Anàlisi dels multiplicadors	21
3.1 Multiplicadors aproximats de 8 bits	21
3.1.1 Versions basades en AM1	21
3.1.2 Versions basades en AM2	23
3.1.3 Versions basades en AM3	23
3.1.4 Versions basades en AM4	24
3.1.5 Comparació dels multiplicadors aproximats de 8 bits	24
3.2 Multiplicadors aproximats de 16 bits	30
3.2.1 Versions dels multiplicadors de 16 bits	30
3.2.2 Comparació dels multiplicadors aproximats de 16 bits	31
3.3 Multiplicadors aproximats de 32 bits	33
3.3.1 Versions dels multiplicadors de 32 bits	34
3.3.2 Comparació dels multiplicadors aproximats de 32 bits	34

4 Planificació temporal	37
4.1 Descripció de les tasques	37
4.1.1 Gestió del projecte (Fita inicial)	37
4.1.2 Configuració de les eines de treball	38
4.1.3 Estudi i implementació del circuit	38
4.1.4 Depuració i simulacions de la implementació	38
4.1.5 Anàlisi dels resultats	38
4.1.6 Comparació dels resultats	38
4.1.7 Redacció de la documentació del projecte	39
4.1.8 Reunions de control i seguiment	39
4.1.9 Fita final	39
4.2 Duració aproximada de les tasques	39
4.3 Gantt	40
4.4 Recursos	40
4.4.1 Software	40
4.4.2 Materials	41
4.4.3 Humans	41
4.5 Gestió del risc: Plans alternatius i obstacles	41
4.6 Desviacions	42
4.6.1 Planificació definitiva	42
5 Gestió econòmica	44
5.1 Pressupost	44
5.1.1 Costos de personal per activitat	44
5.1.2 Costos imputats genèricament	45
5.1.3 Costos de contingència	46
5.1.4 Costos dels imprevistos	46
5.1.5 Resum pressupost	46
5.2 Control de Gestió	47
6 Sostenibilitat i compromís social	48
6.1 Dimensió econòmica	48
6.2 Dimensió social	49
6.3 Dimensió ambiental	50
7 Conclusions	52
8 Referències	53

Índex de figures

1	Esquema Wallace tree 8x8.	15
2	Exemple de multiplicació aproximada amb el multiplicador AM1.	16
3	Mètode de creació de multiplicadors de més bits.	18
4	Esquema AM3 de 4 bits.	19
5	Circuit AM1 part aproximada.	21
6	Circuit AM1.	22
7	Circuit de la unitat de control del AM1_2.	22
8	Circuit AM1_2.	22
9	Unitat de control del multiplicador AM1_3.	23
10	Comparació potència i error entre AM1 i AM4.	27
11	Comparació d'àrea i error entre AM1 i AM4.	27
12	Comparació potència i error entre AM2 i AM4.	28
13	Comparació d'àrea i error entre AM2 i AM4.	28
14	Comparació potència i error dels multiplicadors de 16 bits.	32
15	Comparació d'àrea i error dels multiplicadors de 16 bits.	32
16	Comparació potència i error dels multiplicadors de 32 bits.	35
17	Comparació d'àrea i error dels multiplicadors de 32 bits.	36
18	Diagrama de Gantt.	41
19	Diagrama de Gantt definitiu.	44

Índex de taules

1	Mapa Karnaugh multiplicador exacte.	17
2	Mapa Karnaugh multiplicador AM2.	17
3	Resum de les versions AM2 de 8 bits.	23
4	Resum de les versions AM3 de 8 bits.	24
5	Resum de les versions AM4 de 8 bits.	24
6	Comparació dels multiplicadors AM1, AM2 i AM3 de 8 bits.	25
7	Comparació dels multiplicadors AM1, AM2 i AM4.	26
8	Resum de l'anàlisi dels millors multiplicadors aproximats de 8 bits.	28
9	Resum de les versions dels millors multiplicadors aproximats de 8 bits.	29
10	Resum de les versions de 16 bits.	30
11	Comparació dels multiplicadors de 16 bits.	31
12	Resum de l'anàlisi dels millors multiplicadors de 16 bits.	33
13	Resum de les versions dels millors multiplicadors aproximats de 8 bits.	33
14	Resum de les versions de 32 bits.	34
15	Comparació dels multiplicadors de 32 bits.	35
16	Resum de l'anàlisi dels millors multiplicadors de 16 bits.	36
17	Resum de les versions dels millors multiplicadors aproximats de 8 bits.	36
18	Durada de les tasques.	40
19	Costos de personal per activitat.	45
20	Costos imputats genèricament.	46
21	Costos de contingència.	47
22	Costos dels imprevistos.	47
23	Resum del pressupost.	47

1 Contextualització i abast del projecte

1.1 Context i formulació del problema

Actualment els multiplicadors són una de les unitats funcionals que requereixen més consum d'energia, degut al gran nombre de portes lògiques que contenen. Aquest factor és molt important en dispositius de baix consum que requereixen una bateria com a font d'energia, com podrien ser ordinadors portàtils, telèfons mòbils, sistemes encastats... ja que si reduïm el consum de les seves unitats funcionals més costoses, aconseguirem que la bateria tingui una durada superior. Per aquest motiu, des de fa temps s'estan analitzant diverses tècniques amb la finalitat de reduir-ne el seu consum.

En aquest projecte ens centrarem en la tècnica que s'anomena aproximació computacional [1]. Aquesta tècnica es basa en aprofitar el fet que diverses aplicacions poden tolerar un cert error en el resultat del càlcul de les multiplicacions. Principalment s'utilitza en aplicacions relacionades amb els sentits de l'ésser humà, com podria ser la visió, degut a que els nostres sentits no són perfectes i un petit error en el resultat pot passar desapercebut [2].

Aquesta tècnica es pot aplicar en aplicacions que actualment estan molt esteses, com podria ser en processament de gràfics, reconeixements facials, mineria de dades... per tant, reduir el seu consum pot afectar positivament en molts àmbits.

En l'actualitat, existeixen tres tècniques d'aproximació computacional aplicades als multiplicadors:

- Aproximació en la generació dels productes parcials [3]: consisteix en ometre la generació d'alguns productes parcials i així reduir-ne el nombre que s'han de sumar. D'aquesta manera es redueix l'àrea del circuit, la profunditat de l'arbre d'acumulacions i degut això, es redueix el seu consum d'energia.
- Aproximació en l'arbre de productes parcials [4]: consisteix en dividir els bits de l'entrada en dues parts. Una d'aquestes parts es calcularà de manera exacta, mentre que l'altra es calcularà de manera aproximada mitjançant un circuit que permet aproximar el resultat de la multiplicació.
- Aproximació en la suma de productes parcials [5]: consisteix en utilitzar sumadors aproximats en el circuit del multiplicador. Aquesta estratègia ofereix un nombre molt elevat de combinacions, ja que existeixen diversos tipus de sumadors aproximats i s'ha d'analitzar l'efecte de combinar-los en un mateix multiplicador.

En aquest projecte hem decidit centrar-nos en les dues primeres tècniques. S'ha descartat la tercera, ja que en el fons analitza sumadors aproximats dins d'un multiplicador i no l'estructura en si del multiplicador (com si fan les dues primeres opcions).

El projecte consisteix en implementar en VHDL (Very High Speed Integrated Circuit Hardware Description Language) i analitzar, mitjançant simulacions, diverses implementacions de multiplicadors aproximats. Posteriorment es realitzarà una comparació dels resultats obtinguts, per tal de veure quines són les implementacions més òptimes, tenint en compte un seguit de mètriques.

S'ha triat realitzar aquest projecte per dos motius. Primer, per tal d'aprofundir més en els conceptes apresos en les assignatures Very Large Scale Integration (VLSI) i Projecte d'Enginyeria de Computadors (PEC) del grau d'enginyeria informàtica i concretament, de l'especialitat d'Enginyeria de Computadors. I segon, realitzar un projecte relacionat amb la Sostenibilitat i Compromís Social, ja que la FIB hi està estretament lligada i ha estat una de les competències transversals al llarg de la carrera.

1.2 Actors implicats

A continuació es detallen els actors implicats en el projecte, ja sigui directament o indirectament, i les persones o organitzacions que es poden beneficiar dels resultats d'aquest projecte.

1.2.1 Desenvolupador

És el principal encarregat del projecte. S'encarrega tant de la part de gestió com de la part tècnica. Gestiona, desenvolupa i finalment documenta tot el projecte en una memòria.

El desenvolupador és supervisat pel director del projecte.

1.2.2 Director

El paper del director consisteix en supervisar i donar ajut durant el plantejament i desenvolupament del projecte. Pot ajudar en la resolució d'errors o obstacles que poden sorgir, per tal de poder assolir els objectius marcats. En aquest cas, el director del projecte és en Ramon Canal Corretger del Departament d'Arquitectura de Computadors (DAC) i professor de l'assignatura VLSI en el grau d'enginyeria informàtica.

1.2.3 Beneficiaris

Com que es tracta d'un projecte de recerca, no hi ha un usuari final específic que es beneficiï directament dels resultats finals.

No obstant, pot beneficiar indirectament a fabricants del sector que vulguin aplicar les conclusions obtingudes. En el projecte es determinarà la millor implementació de

multiplicador aproximat, dins de les estudiades. Els fabricants poden utilitzar aquesta implementació en els seus dissenys, si ho consideren beneficiós respecte la seva implementació actual.

També pot beneficiar a investigadors d'aquest sector, ja que poden tenir en compte els resultats obtinguts per estudiar altres implementacions i comparar-les amb les estudiades en aquest projecte. En concret, pot beneficiar al “*DRAC: Designing RISC-V-based Accelerators for next generation Computers*”, projecte en el qual hi està implicat el director d'aquest treball, en Ramon Canal Corretger. Està finançat per la Generalitat de Catalunya dins del programa RIS3CAT i el coordina el BSC. També hi participen la UPC, UB, UAB i URV. Té com a objectiu principal, preparar la base del 2n processador europeu (European Processor Initiative) i una de les alternatives avaluades és la introducció d'unitats aproximades. En aquest projecte s'ha fet un anàlisi previ de les oportunitats de reducció de consum que ofereix l'*approximate computing*.

1.3 Justificació

Es compara el projecte proposat, amb altres estudis similars realitzats prèviament, per tal de justificar la necessitat de desenvolupar aquest projecte.

1.3.1 Estudis similars

A continuació s'expliquen els diversos estudis similars al proposat i s'expliquen les diferències.

El primer estudi [6], correspon a una comparació entre algunes de les possibles implementacions de multiplicadors aproximats. Aquest projecte es centra exclusivament en una de les tècniques d'aproximació de multiplicadors, en concret en la tècnica d'aproximació en la suma de productes parcials, explicada anteriorment. Com ja s'ha comentat, aquest projecte estudiarà les altres dues tècniques d'aproximació.

El segon estudi [7] exposa diferents tipus de multiplicadors aproximats i els explica breument, però no realitza una comparació entre ells per tal de determinar quin és el més òptim.

Degut a què no s'ha trobat un estudi previ que compari en profunditat multiplicadors aproximats, mitjançant les tècniques d'aproximació en la generació dels productes parcials i d'aproximació en l'arbre de productes parcials, podem concloure que es justifica la necessitat per a realitzar aquest projecte.

1.4 Abast del projecte

1.4.1 Objectius

- Estudiar i implementar diverses implementacions de multiplicadors aproximats bassades en l'aproximació en la generació dels productes parcials i en l'aproximació en l'arbre de productes parcials.
- Simular les implementacions per tal de determinar l'error introduït, l'àrea i el consum.
- Analitzar l'efecte de combinar diferents implementacions de multiplicadors aproximats en un mateix disseny.
- Analitzar els resultats obtinguts per tal de determinar quina o quines de les implementacions són les més òptimes.
- Analitzar l'efecte d'ampliar el nombre de bits del multiplicador.

1.4.2 Possibles obstacles i riscos

A continuació es descriuen els possibles obstacles i riscos que es poden produir durant el desenvolupament del projecte i s'expliquen les solucions o plans alternatius que es duren a terme en cas que s'acabin produint.

- Temps: Com que es disposa d'un temps limitat, qualsevol error en la planificació o durant el desenvolupament pot provocar que no es puguin arribar a estudiar algunes de les possibles implementacions. Tot i això, s'arribaran a assolir els objectius mínims marcats i en tot cas seria possible una futura ampliació del projecte incorporant noves implementacions de multiplicadors aproximats.
- Errors en les implementacions: Algunes implementacions són més complexes que altres, tant a l'hora d'implementar com a l'hora de depurar i analitzar. És per això que si a l'hora de depurar no es troba cap manera de resoldre els errors d'una certa implementació en un temps raonable, s'acabarà descartant per tal de no provocar un retard significatiu en les altres tasques del projecte i que podria provocar que no s'arribés a finalitzar.
- Errors en les simulacions: Per obtenir resultats significatius s'haurà de simular cada implementació un gran nombre de vegades. Per aquest motiu, si es produeix un error durant la simulació podria arribar a afectar a la planificació del projecte i fer que no es puguin estudiar implementacions addicionals, reforçant altra vegada l'obstacle del temps limitat.

1.5 Metodologia i rigor

Per a realitzar el projecte s'ha escollit una metodologia de treball incremental o iterativa. Aquesta metodologia consisteix en un conjunt de tasques, anomenades iteració, que es repeteixen.

A continuació s'expliquen les tasques que es duran a terme en cada iteració.

1. Primer de tot s'estudiarà la implementació proposada per tal de comprendre el seu funcionament i implementació.
2. Un cop s'hagi estudiat el seu funcionament, es procedirà a implementar aquest disseny en VHDL amb l'eina de software Quartus II que ens permet implementar, sintetitzar i analitzar circuits.
3. Un cop implementat es faran simulacions, amb l'eina de software Modelsim, per tal de determinar si la implementació realitzada en el pas anterior, funciona de la manera esperada. En cas que no funcioni s'haurà de depurar el circuit, per tal de trobar quin és l'error i poder resoldre'l. Si la implementació funciona correctament, es procedirà a analitzar diferents mètriques, en aquest cas, l'error introduït, l'àrea i el consum d'energia.
4. Un cop realitzats tots aquest passos es procedirà a la següent iteració.

Finalment, quan s'hagin estudiat totes les implementacions possibles dins del temps disponible, es farà un estudi comparatiu entre elles amb els resultats obtinguts, per tal de determinar quina o quines implementacions són les més efectives, tenint en compte les diferents mètriques analitzades.

1.5.1 Eines de treball

El projecte es desenvoluparà utilitzant el sistema operatiu Windows 10 i s'utilitzaran dues eines de treball: el Quartus II i el Modelsim.

- El Quartus II és una eina de software de l'empresa Altera, que forma part d'Intel. Aquesta eina ens permet desenvolupar circuits en diferents llenguatges de programació. També ens permet visualitzar el circuit resultant un cop compilat el disseny. Per a aquest projecte s'ha triat utilitzar aquesta eina i el llenguatge VHDL, ja que s'havien utilitzat prèviament durant l'assignatura Projecte d'Enginyeria de Computadors (PEC) i per tant, no caldrà familiaritzar-se amb eines noves de programació de circuits i així aprofitar millor el temps disponible i també perquè és una eina de software gratuïta.

- El Modelsim és una eina de software de l'empresa Intel, ens serveix per a simular els circuits implementats amb el Quartus II, d'aquesta manera podrem comprovar el correcte funcionament dels multiplicadors implementats i posteriorment analitzar les diferents mètriques. Aquesta eina també va ser utilitzada per a simular circuits en l'assignatura Projecte d'Enginyeria de Computadors (PEC), per tant tampoc caldrà destinar un cert temps per a familiaritzar-se amb eines noves de simulació de circuits i també és una eina de software gratuïta.

Ambdues eines proporcionen una versió gratuïta que es pot descarregar a través de les seves pàgines web [8][9].

1.5.2 Eines per monitoritzar l'evolució del projecte

Per tal de monitoritzar el projecte es disposarà d'una carpeta compartida amb el director al Google Drive [10] on s'aniran penjant les diferents implementacions i documentació associada, per tal de tenir una còpia online de tots els fitxers. S'ha triat aquesta eina ja que és gratuïta i permet crear, editar, de manera online, documents de text i compartir fitxers de tot tipus de format.

També s'utilitzarà Ganttter [11], aquesta és una eina online gratuïta que permet planificar projectes, definint i ordenant les diferents tasques que s'han de dur a terme durant el seu desenvolupament.

1.5.3 Mètodes de validació

Com ja s'ha indicat en l'apartat "*1.5 Metodologia i rigor*", per tal de verificar els resultats es faran simulacions amb l'eina ModelSim i es compararan amb els resultats que s'esperen d'aquell multiplicador aproximat en concret.

Les reunions periòdiques amb el director del projecte també serviran per anar validant els resultats obtinguts i suggerir canvis o millores, si es considera necessari.

2 Estudi dels multiplicadors aproximats

En aquest apartat es desenvolupa la part tècnica del projecte. S'explica el procés que s'ha dut a terme per a implementar i avaluar les diferents implementacions, s'explica la implementació de cada multiplicador aproximat i s'analitzen els resultats obtinguts en les simulacions i anàlisi de consum.

2.1 Procés d'implementació i avaluació dels multiplicadors

Per a implementar tots els dissenys s'ha utilitzat el Quartus Prime 18.1. En el Quartus s'han creat els projectes i s'ha decidit utilitzar la FPGA Cyclone IV E: EP4CE115F29C7 que opera a un voltatge de 1,2V. Per a fer la comparació s'hauria pogut triar qualsevol altre placa, ja que la diferència relativa entre les implementacions hauria de ser molt similar. S'ha triat aquesta, ja que les plaques més noves no oferien la possibilitat de fer el tipus de simulacions necessaris per a fer les estimacions de consum, amb la versió del programa de la que disposem.

Per a realitzar l'estudi d'error introduït i de consum s'ha seguit la Guia d'Usuari: Anàlisi de consum i optimitzacions d'Intel [12]. A continuació s'explica breument el procés que s'ha realitzat.

Primer de tot s'ha d'implementar el multiplicador en el Quartus, un cop implementat ens permet sintetitzar el circuit i generar tres fitxers que s'utilitzaran en el Modelsim per a simular el circuit i posteriorment analitzar el seu consum. Aquest tres fitxers són: Standard Delay Format (SDF) que conté informació sobre els retards del circuit, VHDL Output (VHO) que conté informació sobre el circuit sintetitzat i *Value Change Dump Script* que és un script que ens permetrà guardar els canvis de totes les senyals del multiplicador durant la seva simulació.

Cal destacar que es generen 3 fitxers SDF i VHO:

- 1200mv_0c_slow: correspon a un voltatge de 1,2V, 0°C i model slow.
- 1200mv_85c_slow: correspon a un voltatge de 1,2V, 85°C i model slow. Aquest fitxer és el que s'ha utilitzat per a fer l'anàlisi de consum, ja que són les pitjors condicions de funcionament i, per tant, quan la implementació arriba al seu consum màxim.
- 1200mv_0c_fast: correspon a un voltatge de 1,2V, 85°C i model slow.

Un cop s'han obtingut aquest tres fitxers de la implementació, es crea un projecte en el Modelsim i s'adjunten aquests i els fitxers VHDL del multiplicador. També és necessari utilitzar un *testbench*, aquest és un circuit que ens permet generar les entrades per tal de simular la implementació del multiplicador.

Un cop ha finalitzat la simulació, es genera un fitxer *Value Change Dump* (VCD) que es pot utilitzar en el projecte del Quartus amb l'aplicació Power Analyzer i obtenir una estimació del consum del multiplicador. També podem obtenir estadístiques de l'àrea i freqüència màxima del circuit analitzant el *Report* del Quartus un cop s'ha sintetitzat el circuit.

Per a l'anàlisi de la freqüència màxima de treball el Quartus reporta la freqüència pels models de 1,2V 0°C i 1,2V 85°C, hem triat el primer model ja que són les condicions més favorables, on s'arriba a la freqüència màxima possible del multiplicador.

Com que analitzar manualment un a un cada resultat és inviable, ja que amb 2 operands de 8 bits ja existeixen 65536 combinacions possibles, per a comprovar i analitzar el resultat de cada multiplicador s'ha programat un codi en C++ que comprova la sortida de la simulació del Modelsim. Aquest programa calcula el nombre d'errors, la probabilitat d'error, l'error absolut mitjà, l'error absolut màxim, l'error relatiu mitjà i l'error relatiu màxim.

2.2 Wallace tree

Com a punt de partida, s'ha utilitzat el multiplicador Wallace tree. Aquest multiplicador és un dels multiplicadors exactes amb un consum inferior a les altres alternatives, per tant té sentit que comparem les nostres implementacions amb aquest.

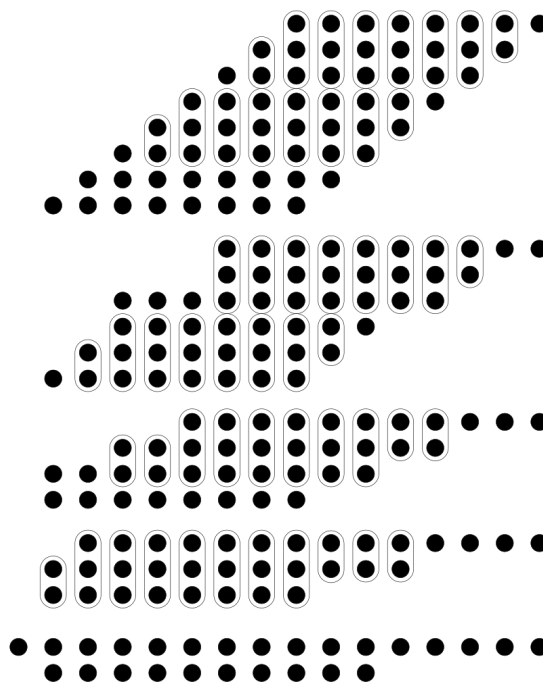


Figura 1. Esquema Wallace tree 8x8.

Font: Katariarra - Own work, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=79466759>

El seu funcionament és molt similar al que s'utilitza per a calcular multiplicacions manualment. Primer de tot, es generen els productes parcials multiplicant cada un dels bits

d'un operand per tots els bits de l'altre operand. Un cop generats els productes parcials (Pas 1) s'han de sumar aquests (Pas 2). Per a fer-ho, s'utilitzen Full-adders (FA) i Half-adders (HA) per a sumar grups de 3 o 2 bits respectivament. Aquest pas s'ha de repetir les vegades necessàries fins a que només quedin dues files de productes parcials. Per últim (Suma final), per a obtenir el resultat de la multiplicació només queda sumar les dues files restants.

2.3 AM1

En aquesta implementació [13] es divideixen els dos operands en dues parts. Aquestes dues parts no tenen perquè ser simètriques però en aquest estudi hem decidit estudiar únicament l'opció on les dues parts tenen la mateixa mida

La part de l'esquerra, que correspon als bits de més pes i que per tant tindran també més pes en el resultat de la multiplicació, es calcula de manera exacte mitjançant un multiplicador Wallace Tree exacte mentre que la part de la dreta, que correspon amb els bits de menys pes, es calcula mitjançant una aproximació.

El mètode d'aproximació és el següent: començant pels bits de més pes de la part aproximada es fa una OR entre els dos bits d'igual pes per tal de veure si algun dels dos és '1', quan es troba un bit a '1', tots els bits restants a la seva dreta es posen a '1'.

En l'exemple de la *Figura 2* es detecta un bit a '1' en la posició 1 començant per l'esquerra, per tant, en el resultat a partir del bit 1 tots els bits seran '1'.

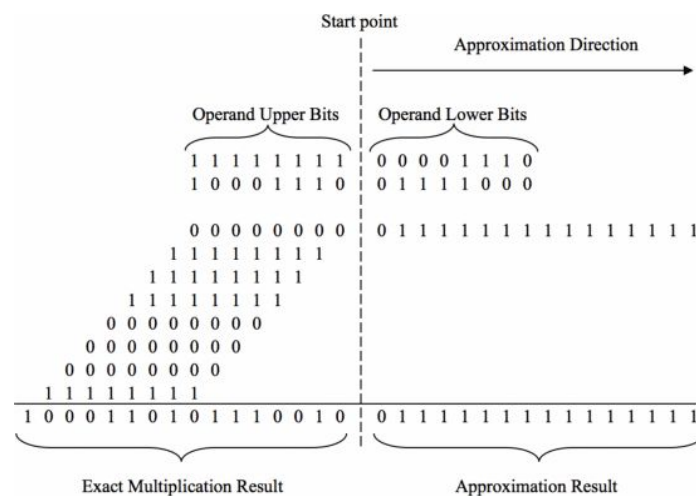


Figura 2. Exemple de multiplicació aproximada amb el multiplicador AM1.

Font: A. Sultan, A. H. Hassan and H. Mostafa, "A Compact Low-Power Mitchell-Based Error Tolerant Multiplier," *2018 New Generation of CAS (NGCAS)*, Villetta, 2018, pp. 130-133.

$A = 65294, B = 36472, A \times B \text{ (real)} = 2381402768, A \times B \text{ (aproximada)} = 2373091327$

Error absolut = valor real - valor obtingut = $2381402768 - 2373091327 = 8311411$

Error relatiu = (error absolut / valor real) * 100 = $(8311411 / 2381402768) * 100 = 0,35\%$

En aquest cas en concret el resultat de la multiplicació té un error absolut de 8311411 i un error relatiu de 0,35%

2.4 AM2

Aquesta implementació [14] consisteix en aprofitar que només hi ha una entrada que produeix 4 bits en una multiplicació de 2 bits ($3 * 3 = 9, b_{10} * b_{10} = b_{1001}$).

Si substituïm aquesta combinació per una sortida de 3 bits, tots ells a '1', el circuit resultant utilitza menys portes i només falla en 1/16 dels casos amb un error relatiu màxim de $((9-7)/9) * 100 = 22,2\%$.

		B1,B0			
A1,A0		00	01	11	10
00		000	000	000	000
01		000	001	011	010
11		000	011	1001	110
10		000	010	110	100

Taula 1. Mapa Karnaugh multiplicador exacte.

$A \times B(0) = A_0 \text{ and } B_0$

$A \times B(1) = (A_0 \text{ and } B_1) \text{ xor } (A_1 \text{ and } B_0)$

$A \times B(2) = (A_1 \text{ and } B_1) \text{ xor } ((A_0 \text{ and } B_1) \text{ and } (A_1 \text{ and } B_0))$

$A \times B(3) = (A_1 \text{ and } B_1) \text{ and } ((A_0 \text{ and } B_1) \text{ and } (A_1 \text{ and } B_0))$

		B1,B0			
A1,A0		00	01	11	10
00		000	000	000	000
01		000	001	011	010
11		000	011	111	110
10		000	010	110	100

Taula 2. Mapa Karnaugh multiplicador AM2.

$A \times B(0) = A_0 \text{ and } B_0$
 $A \times B(1) = (A_0 \text{ and } B_1) \text{ or } (A_1 \text{ and } B_0)$
 $A \times B(2) = A_1 \text{ and } B_1$
 $A \times B(3) = 0$

A partir de multiplicadors de la meitat de bits es poden crear multiplicadors de més bits seguint el mètode de la *Figura 3*.

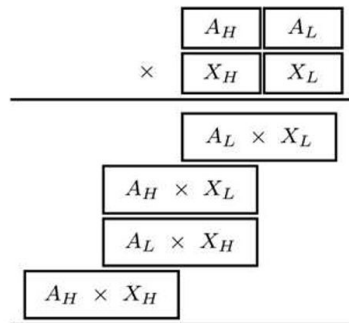


Figura 3. Mètode de creació de multiplicadors de més bits.

Font: P. Kulkarni, P. Gupta and M. Ercegovac, "Trading Accuracy for Power with an Underdesigned Multiplier Architecture," *2011 24th International Conference on VLSI Design*, Chennai, 2011, pp. 346-351.

On $A_L \times X_L$, $A_L \times X_H$, $A_H \times X_L$ i $A_H \times X_H$ correspon al resultat de multiplicadors de la meitat de bits.

A l'hora de fer la suma s'han de shiftar "x" bits els càlculs $A_L \times X_H$ i $A_H \times X_L$, i shiftar "y" bits el càlcul $A_H \times X_H$.

y: correspon al nombre de bits de l'entrada
 x: correspon a la meitat del nombre de bits de l'entrada (y/2).

Exemple:

A(13): 1101, X(6): 0110, A_xX(78): 01001110

ALxXL: 10	ALxXH: 01	AHxXL: 10	AHxXH: 01
x 01	x 01	x 11	x 11
-----	-----	-----	-----
10	01	10	01
+ 00	+ 00	+ 10	+ 01
-----	-----	-----	-----
0010	0001	0110	0011

Suma dels productes:

```

00000010 (ALxXL)
00000100 (ALxXH) (Shift 2)
00011000 (AHxXL) (Shift 2)
+ 00110000 (AHxXH) (Shift 4)
-----
01001110 -> 78

```

2.5 AM3

Aquesta implementació [15] divideix en dues parts els productes parcials generats en el primer pas de l'algoritme Wallace Tree. La part esquerra dels productes parcials es calcula seguint l'algoritme Wallace Tree mitjançant sumadors mentre que la part dreta es calcula mitjançant portes OR de dues i tres entrades.

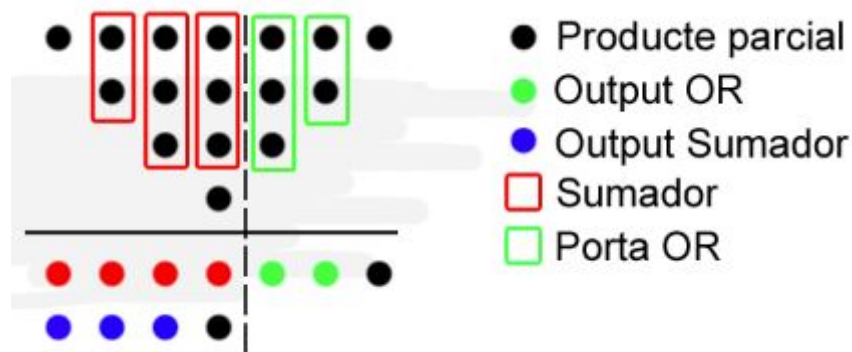


Figura 4. Esquema AM3 de 4 bits.

2.6 AM4

En aquests multiplicadors s'utilitzen combinacions de multiplicadors aproximats entre AM1 i AM3, AM2 i AM3. L'objectiu d'aquestes implementacions és determinar si al combinar diversos multiplicadors aproximats s'obté un consum inferior i determinar si l'error extra introduït és prescindible.

2.7 Mètriques de l'estudi

A continuació s'expliquen les diferents mètriques que s'utilitzaran per tal d'avaluar i comparar el rendiment dels multiplicadors.

- Error: Per a comparar l'error dels multiplicadors es calcula l'error relatiu mitjà i l'error relatiu màxim.

$ERM = \frac{\sum(|\frac{X_o - X}{X} * 100|)}{n}$ on X_o és el resultat obtingut, X és el resultat correcte i n és el nombre de resultats. Pel cas on $X = 0$ el resultat de la divisió seria indefinit. Per aquest cas en concret s'estableix un error del 100%.

$ER_Max = \max(|\frac{X_o - X}{X} * 100|)$ on X_o és el resultat obtingut, X és el resultat correcte i n és el nombre de resultats.

- Àrea: Per a comparar l'àrea dels multiplicadors s'utilitza el nombre de LookUp Tables (LUTs) que ens reporta el Quartus al sintetitzar el disseny. Aquestes són taules de veritat que s'utilitzen per implementar els circuits en FPGAs.

Per a calcular la reducció d'àrea s'utilitza la fórmula:

$R_LUTs = (1 - \frac{nLUT_o}{nLUT}) * 100$ on $nLUT_o$ correspon al nombre de LUTs del disseny que s'està analitzant i $nLUT$ correspon al nombre de LUTs del multiplicador exacte d'aquell nombre de bits.

- Consum: Per a comparar el consum s'utilitza el consum dinàmic de la jerarquia que ens reporta el Quartus. En aquest document ens reporta el consum de cada element del disseny i ens permet veure quins elements són els que consumeixen més, per tal de poder optimitzar-los.

Per a calcular la reducció de consum s'utilitza la fórmula:

$R_PTD = (1 - \frac{PTD_o}{PTD}) * 100$ on PTD_o correspon al consum dinàmic del multiplicador que es vol analitzar i PTD correspon al consum dinàmic del multiplicador exacte d'aquell nombre de bits.

- Freqüència màxima de treball: Per a comparar la freqüència màxima de treball s'utilitza el valor que ens reporta el Quartus al sintetitzar el circuit. Ens reporta dos valors, un que correspon a una temperatura de 0°C i un altre que correspon a una temperatura de 85°C. Per aquesta comparació hem agafat el valor de 0°C ja que correspon amb el valor màxim al que es pot arribar.

3 Anàlisi dels multiplicadors

3.1 Multiplicadors aproximats de 8 bits

En aquest apartat s'expliquen les diferents implementacions realitzades de multiplicadors aproximats de 8 bits. També s'analitza el seu error, consum i àrea i es determina quines de les implementacions són les més òptimes.

3.1.1 Versions basades en AM1

3.1.1.1 AM1_1

En aquest multiplicador es calculen els 4 bits de més pes utilitzant un multiplicador exacte Wallace Tree i els 4 bits de menys pes es calculen mitjançant l'algoritme explicat en l'apartat 2.3.

En la *Figura 5* es mostra el circuit resultant a l'aplicar l'algoritme en la part aproximada de 4 bits i en la *Figura 6* es mostra el circuit complet del AM1.

L'inconvenient que té aquesta implementació és que quan algun dels operands és 0 i l'altre és diferent de 0 en la part baixa, el resultat calculat és diferent de 0 ja que amb aquest algoritme d'aproximació al detectar algun bit a 1 en la part aproximada com a mínim els 5 bits de menor pes seran 1. Posteriorment s'analitzen dues implementacions per a solucionar aquest cas especial de l'algoritme.

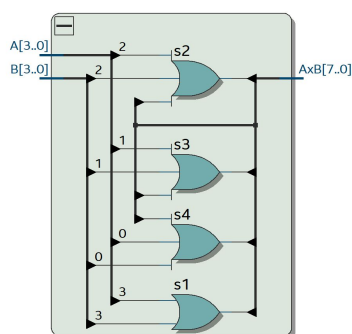


Figura 5. Circuit AM1 part aproximada.

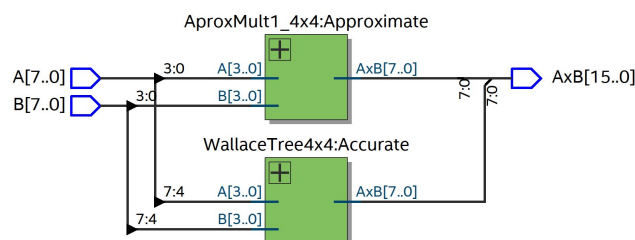


Figura 6. Circuit AM1.

3.1.1.2 AM1_2

En aquesta implementació es solucionen alguns casos on la multiplicació per 0 dóna un resultat diferent de 0. Aquesta implementació correspon exactament amb la implementació presentada en el document *Trading Accuracy for Power with an Underdesigned Multiplier Architecture* [13].

Per a calcular la part dreta de la multiplicació es disposa d'un multiplicador exacte i d'un aproximat (AM1) que calculen el resultat d'aquesta part en paral·lel. També es disposa d'una unitat de control (*Figura 7*) que decideix si s'utilitza el resultat del multiplicador aproximat o exacte. Si la meitat esquerra dels 2 operands val 0, s'utilitza el resultat del multiplicador exacte i en qualsevol altre cas s'utilitza el resultat del multiplicador aproximat. En la *Figura 8* es mostra l'esquema d'aquest multiplicador.

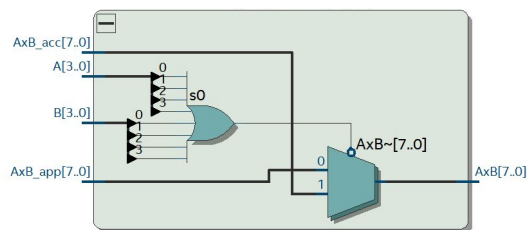


Figura 7. Circuit de la unitat de control del AM1_2

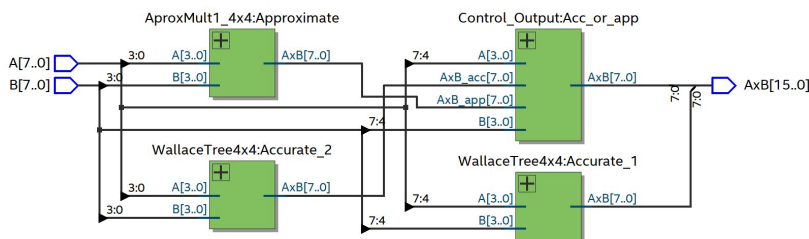


Figura 8. Circuit AM1_2.

3.1.1.3 AM1_3

En aquesta implementació s'acaben de solucionar totes les combinacions errònies on algun dels operands val 0. S'utilitzen els mateixos components que en la versió AM1_2 però canvien les condicions de la unitat de control.

Es distingeixen 3 casos:

- Un o els dos operands són 0: el resultat total de la multiplicació val 0.
- La part dels bits de més pes d'un dels operands és 0: el resultat de la part baixa de la multiplicació correspon al resultat del multiplicador exacte
- En altre cas: el resultat de la part baixa de la multiplicació correspon al resultat del multiplicador aproximat.

La unitat de control del multiplicador AM1_3 (Figura 9) és molt més complexa que la unitat de control del AM1_2 (Figura 7), caldrà veure si l'error que s'ha reduït compensa amb la potència extra que consumirà aquest multiplicador.

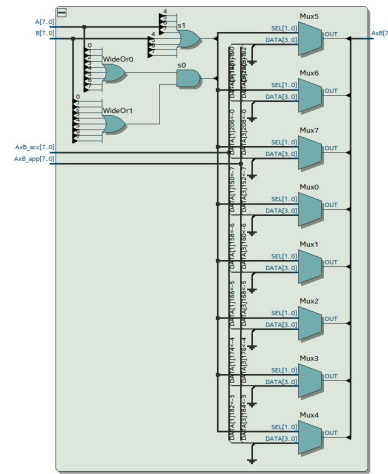


Figura 9. Unitat de control del multiplicador AM1_3.

3.1.2 Versions basades en AM2

S'han implementat 4 versions del multiplicador aproximat 2 (AM2), en cada versió s'ha afegit un multiplicador aproximat extra.

En la taula següent s'indica quins multiplicadors s'han utilitzat en cada versió per a calcular les diferents parts de la multiplicació seguint el mètode de la Figura 3.

Nom	ALxBL	ALxBH	AHxBL	AHxBH
AM2_1	AM2	WT	WT	WT
AM2_2	AM2	AM2	WT	WT
AM2_3	AM2	AM2	AM2	WT
AM2_4	AM2	AM2	AM2	AM2

Taula 3. Resum de les versions AM2 de 8 bits.

3.1.3 Versions basades en AM3

S'han implementat 4 versions del multiplicador aproximat 3 (AM3), en cada versió s'ha afegit un multiplicador aproximat extra com en el cas anterior.

En la Taula 4 s'indica quins multiplicadors s'han utilitzat en cada versió per a calcular les diferents parts de la multiplicació.

Nom	ALxBL	ALxBH	AHxBL	AHxBH
AM3_1	AM3	WT	WT	WT
AM3_2	AM3	AM3	WT	WT
AM3_3	AM3	AM3	AM3	WT
AM3_4	AM3	AM3	AM3	AM3

Taula 4. Resum de les versions AM3 de 8 bits.

3.1.4 Versions basades en AM4

En aquests multiplicadors s'utilitzen combinacions de multiplicadors aproximats entre AM1 i AM3, AM2 i AM3. En la *Taula 5* s'indica quins multiplicadors s'han utilitzat en cada versió per a calcular les diferents parts de la multiplicació.

Nom	ALxBL	ALxBH	AHxBL	AHxBH
AM4_1_1	AM1	-	-	AM3
AM4_1_2_1	AM1_2 + AM3	-	-	WT
AM4_1_2_2	AM1_2 + AM3	-	-	AM3
AM4_1_3_1	AM1_3 + AM3	-	-	WT
AM4_1_3_2	AM1_3 + AM3	-	-	AM3
AM4_2_1_1	AM2	AM3	WT	WT
AM4_2_1_2	AM2	AM3	AM3	WT
AM4_2_1_3	AM2	AM3	AM3	AM3
AM4_2_2_1	AM2	AM2	AM3	WT
AM4_2_2_2	AM2	AM2	AM3	AM3
AM4_2_3	AM2	AM2	AM2	AM3

Taula 5. Resum de les versions AM4 de 8 bits.

3.1.5 Comparació dels multiplicadors aproximats de 8 bits

En la *Taula 6* es mostra una taula resum amb les diferents mètriques analitzades en els multiplicadors aproximats de 8 bits. Per a analitzar aquests multiplicadors s'ha utilitzat un testbench amb totes les combinacions possibles (65536 combinacions).

Llegenda de la taula:

- PTD_B: Consum dinàmic del multiplicador.
- R_PTD_B: Percentatge de reducció del consum.
- ERM: Error relatiu mitjà.
- ER_Max: Error relatiu màxim
- LUTs: Nombre de LUTs (LookUp Tables)
- R_LUTs: Percentatge de reducció de LUTs
- Fmax 0C: Freqüència de treball màxima a 0°C

Nom	PTD_B (mW)	R_PTD_B	ERM	ER_Max	LUTs	R_LUTs	Fmax 0C (MHz)
Wallace Tree	7,05	-	0%	0%	160	-	405,35
AM1_1	0,51	92,76%	22,62%	3088%	36	77,50%	417,54
AM1_2	0,49	93,05%	20,33%	962,5%	76	52,50%	430,85
AM1_3	0,61	91,35%	19,64%	962,5%	82	48,75%	412,20
AM2_1	5,96	15,46%	0,08%	22,22%	147	8,13%	410,68
AM2_2	5,68	19,43%	0,50%	22,22%	134	16,25%	357,27
AM2_3	4,78	32,20%	0,92%	22,22%	121	24,38%	431,03
AM2_4	4,89	30,64%	3,25%	22,22%	108	32,50%	407,50
AM3_1	6,01	14,75%	0,03%	28,57%	152	5,00%	430,48
AM3_2	6,33	10,21%	0,22%	28,57%	144	10,00%	430,66
AM3_3	5,24	25,67%	0,42%	28,57%	136	15,00%	396,20
AM3_4	5,95	15,60%	1,70%	28,57%	128	20,00%	428,63

Taula 6. Comparació dels multiplicadors AM1, AM2 i AM3 de 8 bits.

Analizant els resultats obtinguts podem determinar que les versions més òptimes són les versions AM1_2, AM2_3 i AM3_3 (marcades en color verd) de multiplicadors AM1, AM2 i AM3 respectivament.

A continuació s'analitza l'efecte de combinar multiplicadors AM3 amb multiplicadors AM1 i AM2 tal com s'ha explicat en l'apartat 2.7.4. La taula està ordenada en ordre creixent de consum per a poder comparar-les més fàcilment.

Nom	PTD_B (mW)	ERM	ER_Max	LUTs	R_LUTs
AM4_1_1	0,43	23,91%	3088%	28	82,50%
AM1_2	0,49	22,62%	962,5%	76	52,50%
AM1_1	0,51	20,33%	962,5%	36	77,50%
AM4_1_2_1	0,51	20,33%	962,5%	69	56,88%
AM4_1_2_2	0,53	21,62%	962,5%	61	61,88%
AM4_1_3_2	0,56	20,93%	962,5%	66	58,75%
AM1_3	0,61	19,64%	962,5%	82	48,75%
AM4_1_3_1	0,62	19,65%	962,5%	74	53,75%
AM4_2_1_3	4,76	1,75%	28,57%	123	23,13%
AM2_3	4,78	0,92%	22,22%	121	24,36%
AM2_4	4,89	3,25%	22,22%	108	32,50%
AM4_2_2_2	4,93	1,98%	28,57%	118	26,25%
AM4_2_1_2	4,95	0,46%	28,57%	131	18,13%
AM4_2_2_1	5,09	0,69%	28,57%	126	21,25%
AM4_2_3	5,17	2,20%	28,57%	113	29,28%
AM2_1	5,96	0,08%	22,22%	147	8,13%
AM2_2	5,68	0,50%	22,22%	134	16,25%
AM4_2_1_1	6,04	0,27%	28,57%	139	13,13%

Taula 7. Comparació dels multiplicadors AM1, AM2 i AM4.

Analitzant la *Taula 7* amb els resultats obtinguts de les simulacions dels multiplicadors AM4, observem que algunes versions (marcades en color verd) consumeixen menys que les seves versions anteriors, mentre que d'altres (marcades en color vermell) consumeixen més energia i, per tant, seran descartades.

A continuació es justifica perquè cada versió és o no és viable comparant-la amb la seva versió anterior respectivament:

- AM4_1_1: És viable ja que consumeix menys que la versió AM1_1.
- AM4_1_2_1/AM4_1_2_2: No és viable ja que consumeix més que la versió AM1_2.
- AM4_1_3_1: No és viable ja que consumeix més que la versió AM1_3.
- AM4_1_3_2: És viable ja que consumeix menys que la versió AM1_3.
- AM4_2_1_1: No és viable ja que consumeix més que la versió AM2_1.
- AM4_2_1_2/AM4_2_1_3: És viable ja que consumeix menys que la versió AM2_1.
- AM4_2_2_1/AM4_2_2_2: És viable ja que consumeix menys que la versió AM2_2.
- AM4_2_3: No és viable ja que consumeix més que la versió AM2_1.

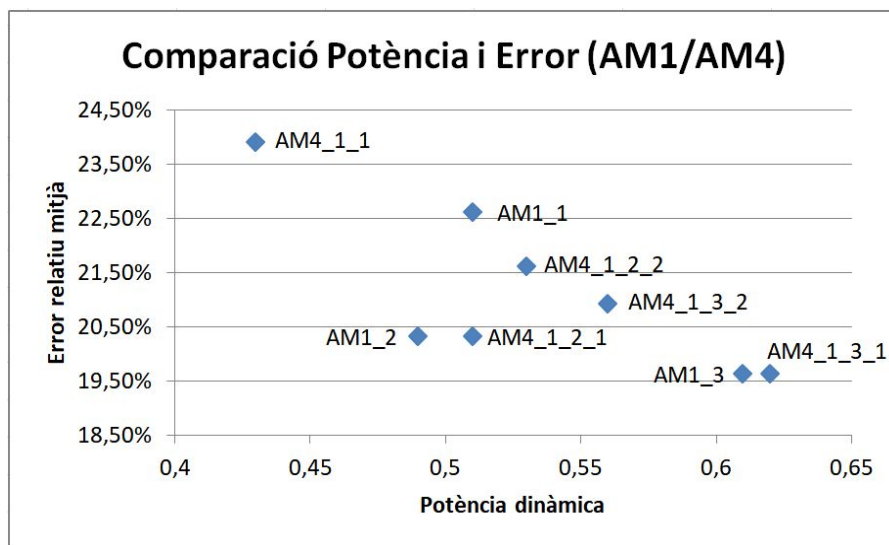


Figura 10. Comparació potència i error entre AM1 i AM4.

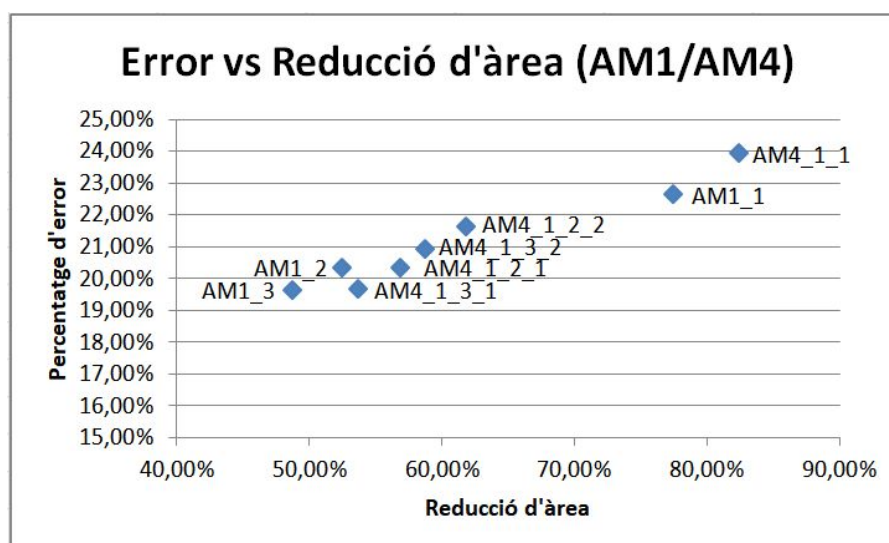


Figura 11. Comparació d'àrea i error entre AM1 i AM4.

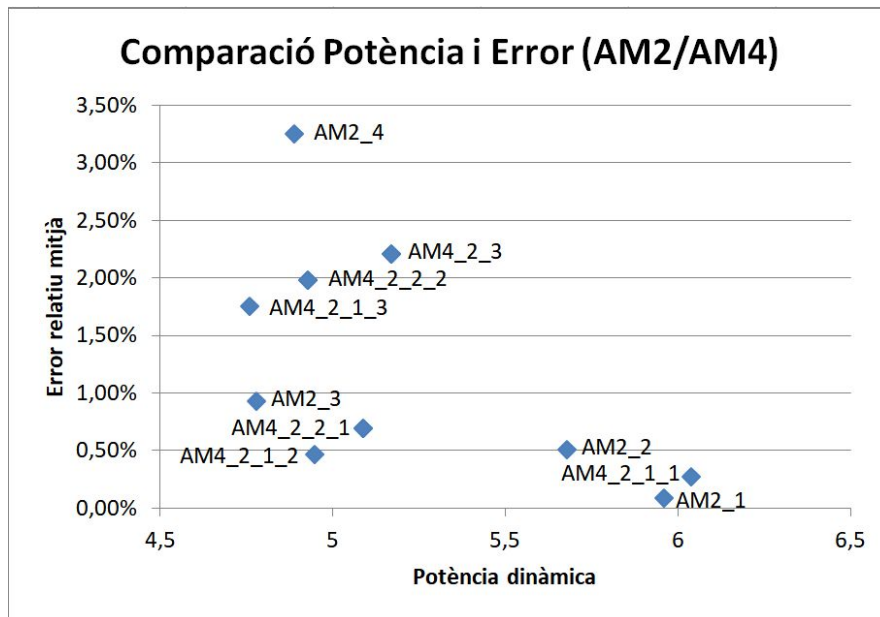


Figura 12. Comparació potència i error entre AM2 i AM4.

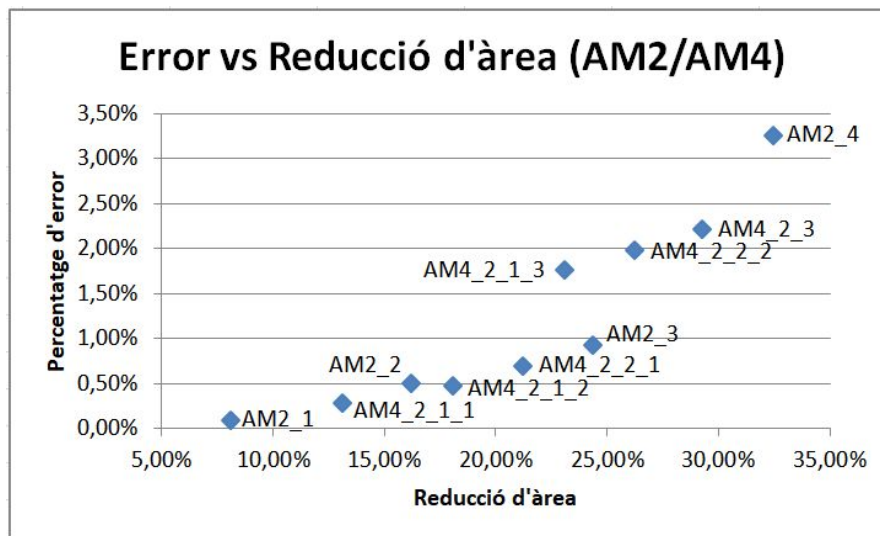


Figura 13. Comparació d'àrea i error entre AM2 i AM4.

Nom	PTD_B (mW)	R_PTT_B	ERM	ER_Max	LUTs	R_A	Fmax 0C (MHz)
AM4_1_1	0,43	93,90%	23,91%	3088%	28	82,50%	430,66
AM4_1_3_2	0,56	92,06%	20,93%	962,5%	66	58,75%	430,48
AM4_2_1_3	4,76	32,48%	1,75%	28,57%	123	23,13%	413,56
AM2_3	4,78	32,20%	0,92%	22,22%	121	24,38%	431,03

Taula 8. Resum de l'anàlisi dels millors multiplicadors aproximats de 8 bits.

Nom	ALxBL	ALxBH	AHxBL	AHxBH
AM4_1_1	AM1	-	-	AM3
AM4_1_3_2	AM1_3 + AM3	-	-	AM3
AM4_2_1_3	AM2	AM3	AM3	AM3
AM2_3	AM2	AM2	AM2	WT

Taula 9. Resum de les versions dels millors multiplicadors aproximats de 8 bits.

En la *Taula 8* es pot trobar un resum del consum, error, àrea i freqüència màxima de treball de les millors versions i en la *Taula 9* està indicat quins tipus de multiplicadors aproximats utilitzen aquestes versions.

Analitzant la comparació de potència i error de la *Figura 10*, la comparació d'error i reducció d'àrea de la *Figura 11* i la taula resum del consum i dels errors relatius de la *Taula 7*, podem concloure que dels multiplicadors basats en la implementació AM1, els més òptims són el AM4_1_1 amb una reducció del consum del 93,90%, una reducció d'àrea del 82,50% però un error relatiu mitjà de 23,91%, i el AM4_1_3_2 amb una reducció del consum del 92,06%, una reducció d'àrea del 58,75% però un error relatiu mitjà de 20,93%. Cal tenir en compte que el AM4_1_1 produeix errors amb multiplicacions per 0 mentre que el AM4_1_3_2 no té aquest inconvenient. S'ha decidit descartar la implementació AM1_2 ja que també produeix errors en la multiplicació per 0 i la versió AM4_1_1 consumeix menys. S'analitzarà una versió que produeix errors amb multiplicacions per 0 (AM4_1_1) i una versió que no produeix errors en aquest cas (AM4_1_3_2). Pel cas de multiplicadors de 8 bits no s'ha estudiat el cas on en la part exacte de la multiplicació s'utilitzés un multiplicador basat en AM2 ja que aquests tenen més error que els AM3 i encara el farien menys viable.

Analitzant la comparació de potència i error de la *Figura 12*, la comparació d'error i reducció d'àrea de la *Figura 13* i la taula resum del consum i dels errors relatius de la *Taula 7*, podem concloure que dels multiplicadors basats en la implementació AM2 els més òptims són el AM4_2_1_3 amb una reducció del consum del 32,48%, una reducció d'àrea del 23,13% i un error relatiu mitjà del 1,75%, i el AM2_3 amb una reducció del consum del 32,20%, una reducció d'àrea del 24,48% i un error relatiu mitjà del 0,92%, el seu consum és pràcticament el mateix mentre que l'error del AM4_2_1_3 és un 0,83% superior. Caldrà veure com és aquesta relació a l'ampliar el nombre de bits del multiplicador.

Com podem observar la reducció del consum dels multiplicadors basats en AM1 és molt significativa però el seu error relatiu mitjà (~20%) és molt superior a l'error relatiu mitjà (~1%) dels multiplicadors basats en AM2, per tant, en multiplicadors aproximats de 8 bits no seria beneficiós utilitzar els multiplicadors AM1 ja que el seu error relatiu és massa gran.

Però caldrà veure el seu comportament al ampliar el nombre de bits ja que l'error relatiu es veurà reduït significativament.

Finalment podem determinar que el multiplicador aproximat AM2_3 és el multiplicador de 8 bits més òptim dins de les implementacions estudiades, ja que consumeix pràcticament el mateix que el AM4_2_1_3 però té un error relatiu mitjà i un error relatiu màxim inferior i també té una freqüència màxima de treball superior.

3.2 Multiplicadors aproximats de 16 bits

En aquest apartat s'estudia l'efecte d'ampliar el nombre de bits del multiplicador a 16 bits. Únicament s'estudien els multiplicadors aproximats que s'han determinat que són més òptims per a 8 bits ja que el temps del que es disposa no és suficient per a analitzar totes les possibles combinacions de multiplicadors, aquests multiplicadors es poden trobar en la *Taula 9*.

3.2.1 Versions dels multiplicadors de 16 bits

En la *Taula 10* s'indiquen els multiplicadors aproximats que s'han utilitzat en cada implementació.

Nom	ALxBL	ALxBH	AHxBL	AHxBH
AM4_1_1	AM1	-	-	AM4_2_1_3
AM4_1_2	AM1	-	-	AM2_3
AM4_2_1	AM1_3+AM4_2_1_3	-	-	AM4_2_1_3
AM4_2_2	AM1_3+AM2_3	-	-	AM2_3
AM4_3_1	AM4_2_1_3	AM4_2_1_3	AM4_2_1_3	AM4_2_1_3
AM4_3_2	AM2_3	AM4_2_1_3	AM4_2_1_3	AM4_2_1_3
AM4_4_1	AM2_3	AM2_3	AM2_3	AM2_3
AM4_4_2	AM2_3	AM2_3	AM2_3	WT

Taula 10. Resum de les versions de 16 bits.

Per a les versions bassades en AM4_1_1 s'ha estudiat l'efecte d'utilitzar un multiplicador AM4_2_1_3 i un AM2_3 en la part exacte de la multiplicació. Per a les versions bassades en AM4_1_3_2 s'ha estudiat l'efecte d'utilitzar tant en la part exacte com en la part aproximada un multiplicador AM2_1_3 i un AM2_3. Per a les versions bassades en AM4_2_1_3 i AM2_3 s'ha estudiat l'efecte d'utilitzar els millors multiplicadors de 8 bits en cada una de les parts de la multiplicació (AM4_3_1 i AM4_4_1) o bé d'utilitzar l'esquema de multiplicadors tal i com ho fan les versions anteriors (AM4_3_2 i AM4_4_2).

3.2.2 Comparació dels multiplicadors aproximats de 16 bits

En la *Taula 11* es mostra una taula resum amb les diferents mètriques analitzades en els multiplicadors aproximats de 16 bits. Per a analitzar aquests multiplicadors s'ha utilitzat un testbench amb 400.000 entrades aleatòries ja que amb 2 operands de 16 bits existeixen 4.294.967.296 combinacions possibles i és inviable analitzar-les totes, com es va fer en el cas de multiplicadors de 8 bits.

Llegenda de la taula:

- PTD_B: Consum dinàmic del multiplicador.
- R_PTD_B: Percentatge de reducció del consum.
- ERM: Error relatiu mitjà.
- ER_Max: Error relatiu màxim
- LUTs: Nombre de LUTs (LookUp Tables)
- R_LUTs: Percentatge de reducció de LUTs
- Fmax 0C: Freqüència de treball màxima a 0°C

Nom	PTD_B (mW)	R_PTD_B	ERM	ER_Max	LUTs	R_A	Fmax 0C (MHz)
Wallace Tree	36,96	-	0%	0%	704	-	299,13
AM4_1_1	10,15	72,54%	4,51%	5281%	131	81,39%	298,15
AM4_1_2	9,90	73,21%	3,70%	5281%	129	81,68%	299,31
AM4_2_1	10,94	70,40%	4,50%	491,4%	282	59,94%	298,78
AM4_2_2	9,80	73,48%	3,68%	491,4%	278	60,51%	280,11
AM4_3_1	35,30	4,49%	1,76%	28,54%	556	21,02%	299,31
AM4_3_2	32,61	11,77%	1,76%	28,54%	554	21,31%	298,15
AM4_4_1	32,86	11,09%	0,92%	22,22%	548	22,16%	297,62
AM4_4_2	33,49	9,39%	0,03%	22,19%	587	16,62%	297,97

Taula 11. Comparació dels multiplicadors de 16 bits.

En color verd estan marcats els multiplicadors aproximats que consumeixen menys entre els multiplicadors AM4_1, AM4_2, AM4_3 i AM4_4 respectivament. Com podem observar en el cas del AM4_3_2 la millors versió ha estat seguint l'esquema de multiplicadors de la

versió de 8 bits (1 AM2 i 3 AM3) en comptes d'utilitzar 4 multiplicadors AM4_2_1_3 mentre que la millor versió del AM4_4 ha estat utilitzant 4 multiplicadors AM2_3. Cal destacar que l'error dels multiplicadors basats en AM1 s'ha reduït significativament. En multiplicadors de 8 bits el seu error relatiu mitjà era de ~20-23% mentre que amb els multiplicadors de 16 bits és de només un ~3-4%

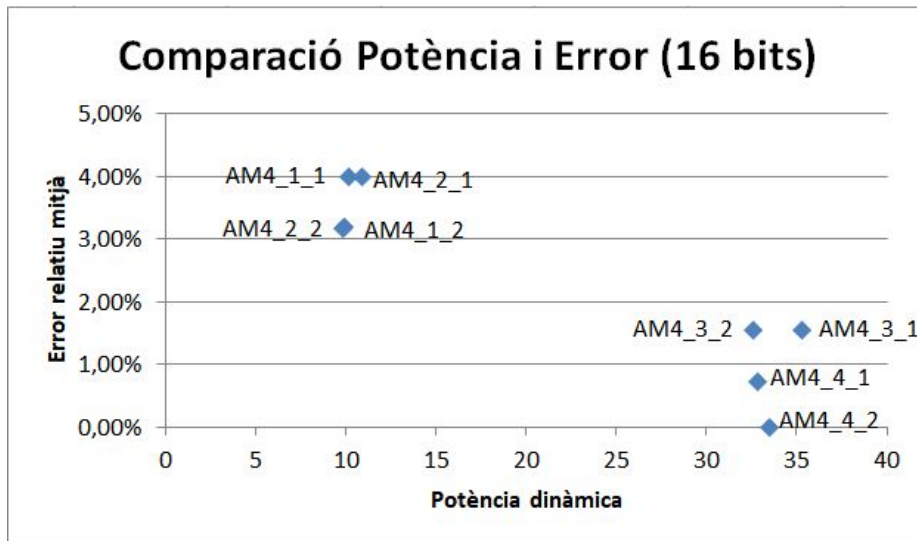


Figura 14. Comparació potència i error dels multiplicadors de 16 bits.

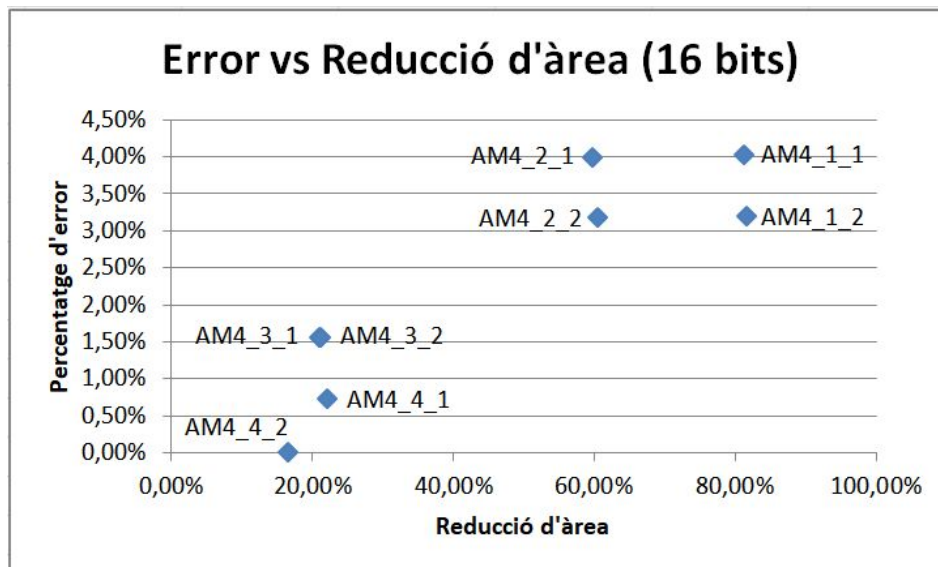


Figura 15. Comparació d'àrea i error dels multiplicadors de 16 bits.

Nom	PTD_B (mW)	R_PTT_B	ERM	ER_Max	LUTs	R_A	Fmax 0C (MHz)
AM4_2_2	9,80	73,48%	3,68%	491,4%	278	60,51%	280,11
AM4_3_2	32,61	11,77%	1,76%	28,54%	554	21,31%	298,15
AM4_4_1	32,86	11,09%	0,92%	22,22%	548	22,16%	297,62
AM4_4_2	33,49	9,39%	0,03%	22,19%	587	16,62%	297,97

Taula 12. Resum de l'anàlisi dels millors multiplicadors de 16 bits.

Nom	ALxBL	ALxBH	AHxBL	AHxBH
AM4_2_2	AM1_3+AM2_3	-	-	AM2_3
AM4_3_2	AM2_3	AM4_2_1_3	AM4_2_1_3	AM4_2_1_3
AM4_4_1	AM2_3	AM2_3	AM2_3	AM2_3
AM4_4_2	AM2_3	AM2_3	AM2_3	WT

Taula 13. Resum de les versions dels millors multiplicadors aproximats de 8 bits.

En la *Taula 12* es pot trobar un resum del consum, error, àrea i freqüència màxima de treball de les millors versions i en la *Taula 13* està indicat quins tipus de multiplicadors aproximats utilitzen aquestes versions.

Analitzant la comparació de potència i error de la *Figura 14*, la comparació d'error i reducció d'àrea de la *Figura 15* i la taula resum de l'anàlisi de la *Taula 11*, podem concloure que dels multiplicadors basats en la implementació AM1, el més òptim és el AM4_2_2 amb una reducció del consum del 73,48%, una reducció d'àrea del 60,51% i un error relatiu mitjà de 3,68%. La versió AM4_1_2 que té un consum similar s'ha descartat ja que aquella versió produeix errors en multiplicacions per 0 mentre que la versió AM4_2_2 no produeix errors en aquest cas, consumeix menys i té un error relatiu inferior.

Dels multiplicadors basats en la implementació AM2, trobem que els més òptims són el AM4_3_2 amb una reducció del consum del 11,77%, una reducció d'àrea del 21,31% i un error relatiu mitjà de 1,76% i el AM4_4_1 amb una reducció del consum del 11,09%, una reducció d'àrea del 22,16% i un error relatiu mitjà de 0,92%. Com que tenen un consum pràcticament idèntic, caldrà tornar a analitzar-los al dos al fer l'estudi de multiplicadors de 32 bits. També cal destacar el AM4_4_2 ja que amb un error del 0,03% aconsegueix una reducció del consum del 9,39% i una reducció d'àrea del 16,62%.

El multiplicador aproximat de 16 bits més òptim és el AM4_2_2 amb una reducció del consum del 73,48% i un error relatiu mitjà de 3,68%. Per a aplicacions que requereixin un nivell de precisió superior al d'aquest multiplicador, la següent millor opció és el multiplicador AM4_4_2 amb una reducció del consum del 9,39% i una reducció d'àrea del 16,62% però amb només un 0,03% d'error relatiu mitjà.

3.3 Multiplicadors aproximats de 32 bits

En aquest apartat s'estudia l'efecte d'ampliar el nombre de bits del multiplicador a 32 bits. Igual que en el cas anterior, únicament s'estudien els multiplicadors aproximats que s'han determinat que són més òptims per a 16 bits, ja que el temps del que es disposa no és suficient per a analitzar totes les possibles combinacions de multiplicadors, aquests multiplicadors es poden trobar en la *Taula 13*.

3.3.1 Versions dels multiplicadors de 32 bits

Nom	ALxBL	ALxBH	AHxBL	AHxBH
AM4_2_2_1	AM1+AM4_4_1	-	-	AM4_4_1
AM4_2_2_2	AM1+AM4_3_2	-	-	AM4_3_2
AM4_3_2	AM4_4_1	AM4_2_1_3	AM4_2_1_3	AM4_2_1_3
AM4_4_1_1	AM4_4_1	AM4_4_1	AM4_4_1	AM4_4_1
AM4_4_2_2	AM4_4_1	AM4_4_1	AM4_4_1	WT

Taula 14. Resum de les versions de 32 bits.

Per a les versions basades en AM4_2_2 s'ha estudiat l'efecte, seguint el mateix esquema que en els multiplicadors de 16 bits, d'utilitzar tant en la part exacte com en la part aproximada un multiplicador AM4_4_1 i un AM4_3_2 ja que són els millors multiplicadors aproximats de 16 bits basats en AM2 i AM3.

Pel que fa als AM4_3_2, AM4_4_1 i AM4_4_2 només s'estudia una versió seguint el mateix esquema dels multiplicadors de 16 bits.

3.3.2 Comparació dels multiplicadors aproximats de 32 bits

En la *Taula 15* es mostra una taula resum amb les diferents mètriques analitzades en els multiplicadors aproximats de 32 bits. Per a analitzar aquests multiplicadors s'ha utilitzat un testbench amb 400.000 entrades aleatòries ja que amb 2 operands de 32 bits és inviable analitzar totes les entrades.

Llegenda de la taula:

- PTD_B: Consum dinàmic del multiplicador.
- R_PTD_B: Percentatge de reducció del consum.
- ERM: Error relatiu mitjà.
- ER_Max: Error relatiu màxim
- LUTs: Nombre de LUTs (LookUp Tables)
- R_LUTs: Percentatge de reducció de LUTs
- Fmax 0C: Freqüència de treball màxima a 0°C

Nom	PTD_B (mW)	R_PTT_B	ERM	ER_Max	LUTs	R_A	Fmax 0C (MHz)
Wallace Tree	116,28	-	0%	0%	2944	-	183,59
AM4_2_2_1	34,29	70,51%	0,95%	100%	1171	60,22%	207,17
AM4_2_2_2	32,83	71,77%	3,84%	100%	1183	59,82%	192,42
AM4_3_2	104,45	10,17%	3,80%	28,55	2338	20,58%	161,37
AM4_4_1_1	107,57	7,49%	0,91%	22,21%	2320	21,20%	163,43
AM4_4_2_2	106,57	8,35%	0,0005%	9,31%	2476	15,90%	172,65

Taula 15. Comparació dels multiplicadors de 32 bits.

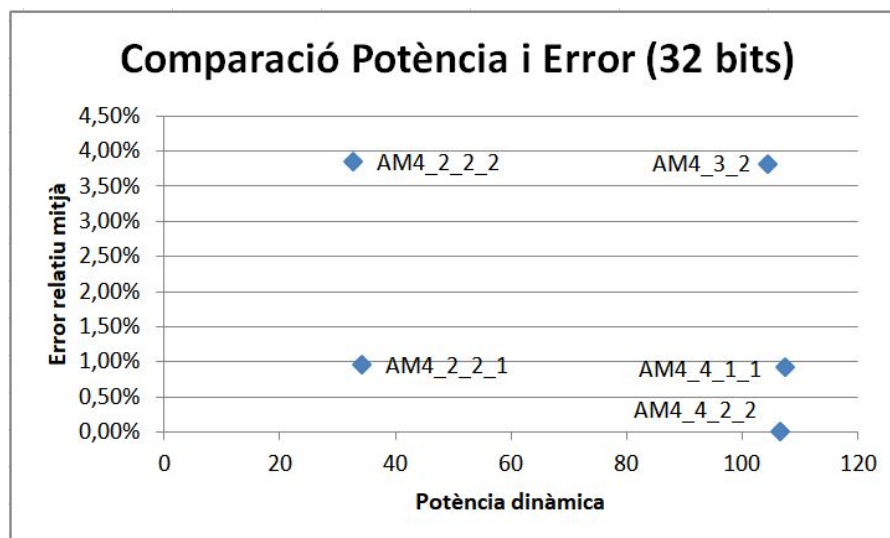


Figura 16. Comparació potència i error dels multiplicadors de 32 bits.

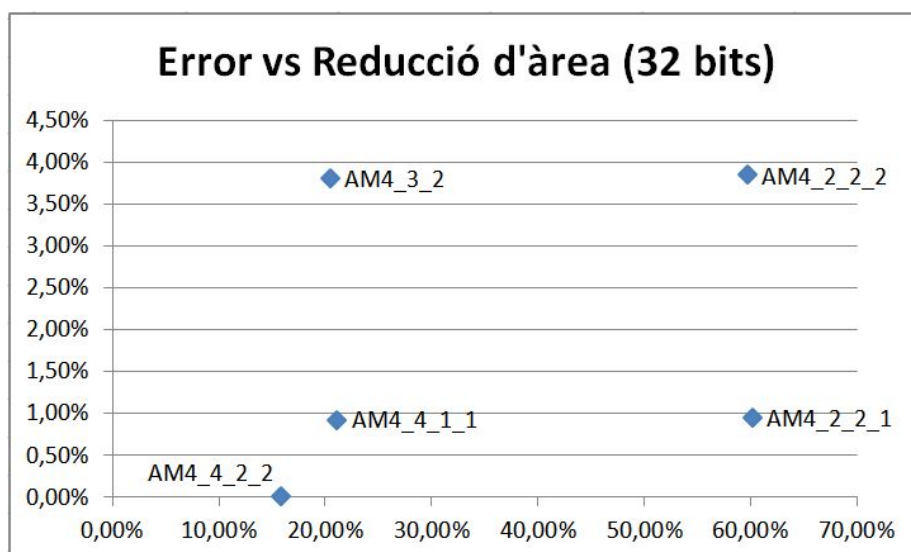


Figura 17. Comparació d'àrea i error dels multiplicadors de 32 bits.

Nom	PTD_B (mW)	R_PTT_B	ERM	ER_Max	LUTs	R_A	Fmax 0C (MHz)
AM4_2_2_1	34,29	70,51%	0,95%	100%	1171	60,22%	207,17
AM4_2_2_2	32,83	71,77%	3,84%	100%	1183	59,82%	192,42
AM4_4_2_2	106,57	8,35%	0,0005%	9,31%	2476	15,90%	172,65

Taula 16. Resum de l'anàlisi dels millors multiplicadors de 16 bits.

Nom	ALxBL	ALxBH	AHxBL	AHxBH
AM4_2_2_1	AM1+AM4_4_1	-	-	AM4_4_1
AM4_2_2_2	AM1+AM4_3_2	-	-	AM4_3_2
AM4_4_2_2	AM4_4_1	AM4_4_1	AM4_4_1	WT

Taula 17. Resum de les versions dels millors multiplicadors aproximats de 8 bits.

En la *Taula 16* es pot trobar un resum del consum, error, àrea i freqüència màxima de treball de les millors versions i en la *Taula 17* està indicat quins tipus de multiplicadors aproximats utilitzen aquestes versions.

Analitzant la comparació de potència i error de la *Figura 16*, la comparació d'error i reducció d'àrea de la *Figura 17* i la taula resum de l'anàlisi de la *Taula 15*, podem concloure que els millors multiplicadors aproximats de 32 bits són el AM4_2_2_1 amb una reducció del consum del 70,51%, una reducció d'àrea del 60,22% i un error relatiu mitjà de 0,95%, amb

una reducció del consum del 71,77%, una reducció d'àrea del 59,82% i un error relatiu mitjà de 3,84%.

Altra vegada cal destacar el AM4_4_2_2 ja que amb un error del 0,0005% aconseguix una reducció del consum del 8,35% i una reducció d'àrea del 15,90%.

Les altres versions queden descartades ja que les versions AM4_2_2_1 i AM4_2_2_2 tenen pràcticament el mateix error que les versions AM4_4_1_1 i AM4_3_2 respectivament però el seu consum és molt significativament inferior.

Finalment, podem determinar que el multiplicador aproximat AM4_2_2_1 és el multiplicador de 32 bits més òptim dins de les implementacions estudiades. Per a aplicacions que requereixin un nivell d'error inferior al 1% es podria utilitzar la versió AM4_2_2_1 i per a aplicacions on es requereixi una precisió quasi perfecte es podria utilitzar el multiplicador AM4_4_2_2.

4 Planificació temporal

En aquest apartat s'expliquen les diverses tasques que es duran a terme durant la realització del projecte, descrivint cada tasca amb detall i especificant la seva durada aproximada. També s'inclouen els recursos humans i materials que seran necessaris.

La durada d'aquest projecte és d'aproximadament 4 mesos, des del 16 de Setembre del 2019 fins al 23 de Gener del 2020. En concret, el projecte dura 129 dies. El projecte consta de 18 crèdits, per tant s'hauria de treballar aproximadament entre 450 i 540 hores (entre 25 i 30 hores per crèdit).

Cal tenir en compte que les duracions que s'indiquen a continuació són una aproximació i es poden veure modificades durant el desenvolupament del projecte.

També cal tenir en compte que amb la metodologia escollida, les tasques 1.1.3, 1.1.4 i 1.1.5 es repeteixen de manera iterativa, és a dir, primer s'estudiarà, s'implementarà i es simularà una implementació i a continuació es realitzaran les mateixes tasques amb la següent implementació.

Es definirà un temps aproximat global per a cada una d'aquestes tasques, però s'ha de tenir en compte que el temps que es destini a cada iteració pot variar significativament degut a la complexitat d'algunes implementacions i als errors trobats al depurar el circuit. És per això que es va triar utilitzar una metodologia incremental, en el temps disponible s'estudiaran el màxim nombre d'implementacions possibles.

4.1 Descripció de les tasques

4.1.1 Gestió del projecte (Fita inicial)

Aquesta tasca consisteix en aprendre a gestionar un projecte i realitzar una sèrie de subtasques que s'expliquen a continuació per tal d'enfocar i planificar el projecte.

- 1) Aprenentatge (14,75h): Consisteix en l'aprenentatge dels diversos mètodes i passos a dur a terme en la gestió d'un projecte.
- 2) Context i abast (24,5h): A partir d'aquesta subtasca es comença la redacció de la memòria del projecte. En concret, es defineix el context, l'abast, els objectius, la metodologia de treball a seguir, els possibles obstacles i es justifica la necessitat de realitzar el projecte.
- 3) Planificació (8,25h): Es realitza una planificació temporal de tot el projecte. Es detallen les tasques a realitzar i es fa una aproximació de les hores que es dedicaran en cada tasca.

- 4) Pressupost i sostenibilitat (9,25h): Es fa un pressupost del projecte i es fa una valoració de l'impacte social i mediambiental que pot tenir el projecte.
- 5) Documentació final (18,25h): Finalment, s'acaba de redactar i polir la memòria del treball realitzada en els passos anteriors, per tal d'arreglar errors i ampliar la documentació, si es considera necessari.

4.1.2 Configuració de les eines de treball

En aquesta tasca s'instal·larà i es configurarà l'entorn de treball. Com que s'utilitzaran eines de desenvolupament i de simulació de circuits que ja s'han utilitzat prèviament, només caldrà dedicar una petita porció de temps per a recordar com funcionen.

4.1.3 Estudi i implementació del circuit

En aquesta fase es decidirà quina implementació es vol estudiar, un cop triada es procedirà a estudiar el seu disseny i implementació per tal de comprendre el seu funcionament. Finalment, s'implementarà el circuit en VHDL utilitzant el programa Quartus II.

S'ha definit un temps global de 120h per a realitzar els diversos estudis i implementacions. Tal i com s'ha explicat anteriorment, és difícil determinar el temps concret que requerirà cada implementació. Considerem que es requereixen entre 20 i 30 hores per a estudiar i implementar cada disseny, per tant, podria ser possible estudiar entre 4 i 6 dissenys, tot i que aquest nombre podria ser superior.

4.1.4 Depuració i simulacions de la implementació

Un cop acabada la implementació, es depurarà el circuit per tal de determinar si la implementació funciona de la forma esperada, o bé, determinar i arreglar els errors que s'hagin pogut produir.

Si el disseny funciona correctament, es passarà a realitzar un cert nombre de simulacions i es guardaran aquest resultat per al seu futur anàlisi. S'haurà de determinar quin nombre de simulacions és adequat per tal d'extreure conclusions fiables en el seu anàlisi.

Per a depurar el programa i realitzar les simulacions s'utilitzarà el programa Modelsim.

4.1.5 Anàlisi dels resultats

En aquesta tasca s'analitzaran els resultats obtinguts en les simulacions del circuit per tal de determinar el valor de les diverses mètriques de l'estudi: l'error induït, l'àrea, el consum i l'energia.

4.1.6 Comparació dels resultats

Un cop s'hagin analitzat totes les implementacions possibles en el temps marcat, es procedirà a comparar els resultats obtinguts en l'anàlisi de les simulacions i s'extrauran conclusions.

4.1.7 Redacció de la documentació del projecte

Al llarg de tot el projecte s'anirà documentant tot el seu desenvolupament i els resultats que es vagin obtenint en cada tasca.

4.1.8 Reunions de control i seguiment

Al llarg de tot el projecte també es faran reunions amb el director del projecte per tal de realitzar un control i seguiment de la feina realitzada .

4.1.9 Fita final

L'última tasca del projecte consisteix en preparar la memòria definitiva que s'entregarà al finalitzar el projecte, revisant que tot el projecte està documentat i és correcte i preparar la defensa que es farà davant del tribunal.

4.2 Duració aproximada de les tasques

En la *Taula 1* s'exposa una taula resum amb la duració aproximada de cada tasca.

Tasca	Duració aproximada en hores
Gestió del projecte (Fita inicial)	75
Configuració de les eines de treball	10
Estudis i implementacions	120
Depuracions i simulacions	95
Anàlisi dels resultats	60
Comparació del les implementacions	30
Redacció de la documentació del projecte	40
Reunions de control i seguiment	15
Fita final	20
Total	465h

Taula 18. Durada de les tasques.

4.3 Gantt

En la *Figura 1* es mostra el diagrama de Gantt obtingut distribuint temporalment les tasques explicades anteriorment amb una dedicació d'unes 30h setmanals.

S'ha considerat que es realitzaran 5 iteracions de les tasques "2.1.3", "2.1.4" i "2.1.5" distribuint les hores de manera equitativa. També es poden observar les dependències entre tasques, marcades amb fletxes.

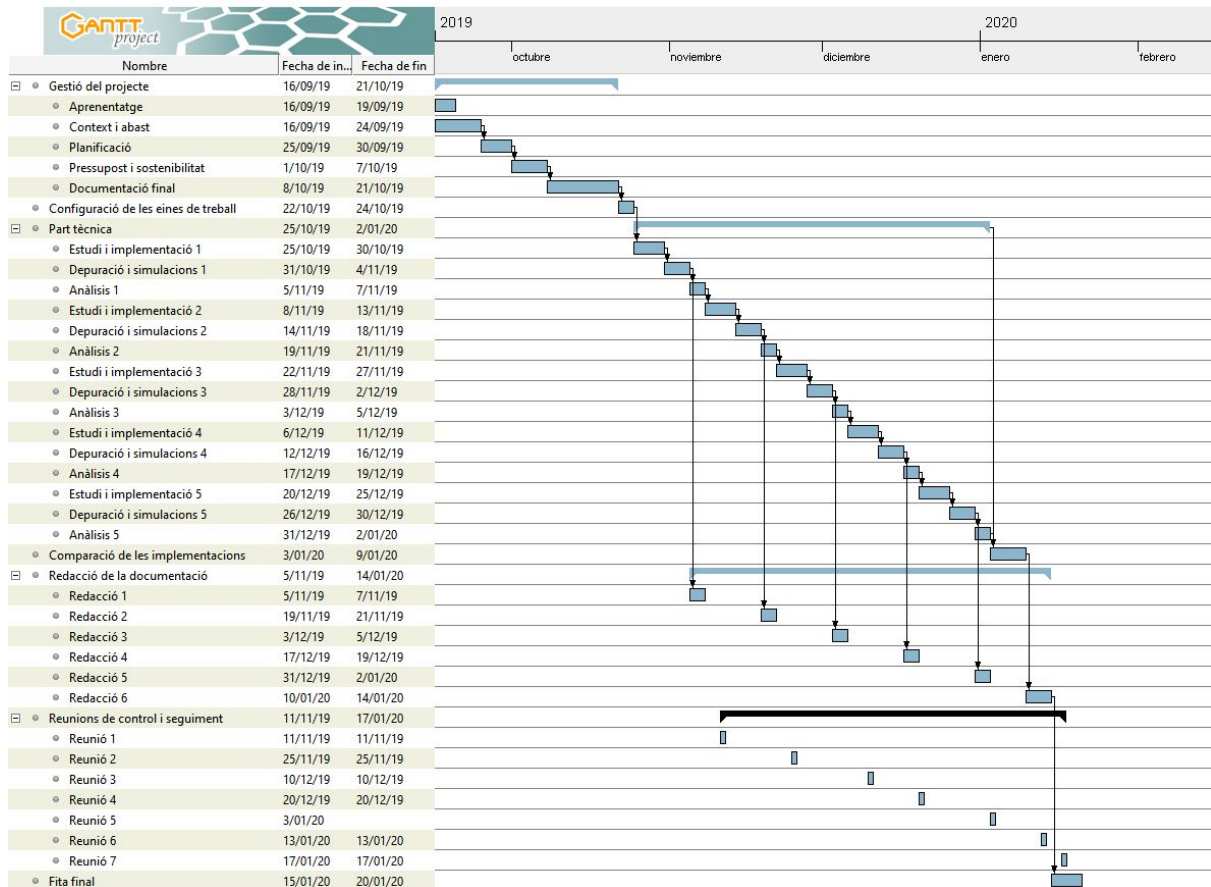


Figura 18. Diagrama de Gantt. Diagrama generat amb el programa GanttProject.

4.4 Recursos

A continuació es descriuen els recursos software, hardware i humans necessaris per a realitzar aquest projecte.

4.4.1 Software

Els recursos de software necessaris són els següents:

- Quartus II: per a programar circuits en VHDL.
- Modelsim: per a simular circuits.
- Microsoft Office: per a redactar la documentació.

- Google drive: per a guardar còpies online de tots els fitxers.
- GanttProject: per a dissenyar el diagrama de Gantt.
- Racó FIB - UPC: per a mantenir contacte amb el director del projecte a través d'emails.

El GanttProject s'utilitzarà en la tasca "*Gestió del projecte*", el Quartus II en la tasca "*Estudis i implementacions*", el ModelSim en la tasca "*Depuracions i simulacions*" i la resta d'eines s'utilitzaran al llarg de totes les tasques del projecte.

4.4.2 Materials

Per a desenvolupar el treball serà necessari disposar d'un lloc físic on poder treballar. Aquest lloc serà principalment la Facultat d'Informàtica de Barcelona, o bé a casa. En el cas de treballar en la facultat, comporta un cost extra que és el de transport.

Per a desenvolupar totes les tasques del projecte serà necessari disposar d'un ordinador personal. En aquest cas es disposarà d'un ordinador portàtil per tal de poder treballar en qualsevol lloc. En aquest ordinador s'instal·laran totes les eines de software necessàries, descrites en l'apartat anterior.

4.4.3 Humans

El projecte serà desenvolupat per una sola persona que tindrà els rols de cap de projecte, desenvolupador i tester. En aquest cas correspon amb l'estudiant i l'hi dedicarà unes 30h setmanals. També es contarà amb la supervisió i ajut del director del projecte.

4.5 Gestió del risc: Plans alternatius i obstacles

Durant el desenvolupament del projecte poden sorgir diversos obstacles tal i com s'explica en l'apartat "*1.4.2 Possibles obstacles i riscos*". Principalment aquests obstacles són els temps i els errors, i poden provocar un retard en la planificació inicial del projecte, per tant, s'han de planificar les possibles solucions a aquestes problemes per si s'arribessin a produir.

Principalment poden sorgir obstacles en les tasques "*Estudis i implementacions*" i "*Depuracions i simulacions*" ja que són les tasques més complexes. Per a superar aquests obstacles primer de tot s'intentarà donar més temps del planificat per a realitzar la tasca en qüestió, això pot provocar que s'hagin de treballar entre 1 i 2 hores diàries per a acabar el projecte a temps. Si es preveu que el temps necessari per a solucionar el problema és excessiu i pot perjudicar la resta d'implementacions del projecte, existeixen dues alternatives: continuar fins a finalitzar la implementació actual i descartar-ne una de futura o bé, en cas extrem, descartar la implementació actual i començar amb la següent implementació. En aquest últim cas, si per algun motiu sobrés temps al final del projecte, s'intentaria acabar la implementació que s'havia descartat.

4.6 Desviacions

Inicialment es va proposar estudiar 5 implementacions, però finalment s'ha decidit fer l'estudi de 3 implementacions de multiplicadors aproximats i després estudiar l'efecte de combinar-los entre ells. S'havia pensat poder arribar a estudiar també multiplicadors de 64 bits si hagués sobrat temps al final del projecte, però aquest no ha estat el cas.

Principalment la planificació s'ha vist afectada en l'etapa de l'anàlisi dels resultats de la primera implementació, degut a que s'ha hagut de fer un codi per tal d'analitzar la sortida de la Simulació del Modelsim i s'ha hagut d'aprendre a fer una estimació del consum del circuit utilitzant el Quartus i el Modelsim.

Aquest imprevist ha provocat un retard d'aproximadament 1 setmana. Contant que aproximadament es treballen 30h setmanals i que el preu per hora teòric és de 27€, l'imprevist suposa una despesa de 810€. El pressupost previst pels imprevistos era de 1161€, per tant, el cost d'aquest imprevist s'ha pogut solucionar amb el pressupost que es va dissenyar.

4.6.1 Planificació definitiva

En la *Figura 15* es presenta el diagrama de Gantt definitiu modificant el nom de les tasques d'estudi de les implementacions i afegint les següents tasques:

- Multiplicadors 16, 32 i 64 bits: correspon a l'estudi i implementació de 16, 32 i/o 64 bits, depenent del temps disponible, de les millors implementacions obtingudes en la tasca anterior Comparació de les implementacions.
- Comparació implementacions 2: correspon a la comparació dels multiplicadors aproximats de 16, 32 i/o 64 bits per tal de determinar l'efecte causat a l'ampliar el nombre de bits.

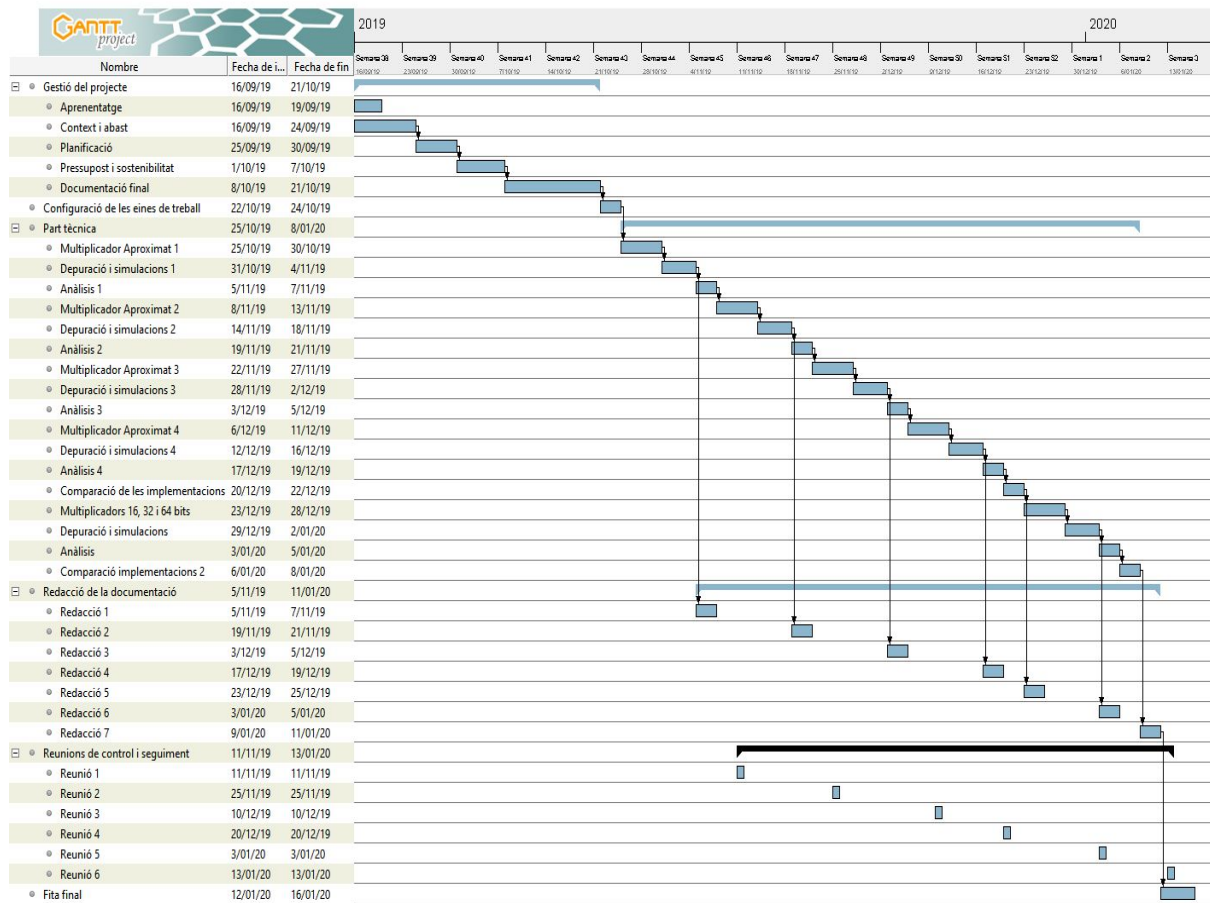


Figura 19. Diagrama de Gantt definitiu. Diagrama generat amb el programa GanttProject.

5 Gestió econòmica

En aquest apartat es presenta el cost aproximat que tindrà la realització del projecte. Cal tenir en compte que l'estudiant no rebrà cap remuneració per haver realitzat el projecte. Tot i això, es fa un pressupost per tal de determinar el cost real que podria tenir.

5.1 Pressupost

5.1.1 Costos de personal per activitat

En la *Taula 19* es mostren els costos de personal per a cada activitat definida en el diagrama de Gantt. S'indiquen les hores destinades a cada activitat, les persones involucrades i el preu resultant total de cada activitat. En aquest cas es consideren dos rols, el de cap de projecte i el de desenvolupador. Consultant els salaris [16], aproximadament el cap de projecte cobra 50€ per hora i el desenvolupador 27€ per hora.

Activitat	Hores	Persones involucrades	Preu total (€)
Gestió del projecte	75	Cap de projecte	3.750
Configuració de les eines de treball	10	Desenvolupador	270
Estudis i implementacions	120	Desenvolupador	3.240
Depuracions i simulacions	95	Desenvolupador	2.565
Anàlisi dels resultats	60	Desenvolupador	1.620
Comparació del les implementacions	30	Desenvolupador	810
Redacció de la documentació del projecte	40	Desenvolupador	1.080
Reunions de control i seguiment	15	Desenvolupador i cap de projecte	1.155
Fita final	20	Desenvolupador	540
Total CPA	465		15.030

Taula 19. Costos de personal per activitat.

5.1.2 Costos imputats genèricament

En la *Taula 20* es mostren els costos del recursos materials que s'utilitzaran al llarg del projecte. S'indiquen les unitats, la vida útil, el nombre de dies que s'utilitzen, el preu per unitat i el preu total o amortització per a cada material.

Per a calcular l'amortització del hardware i software es té en compte el preu, la vida útil i les hores que es farà servir cada recurs.

Per a calcular el cost de l'electricitat es té en compte el preu per hora de l'electricitat en aquest moment i les hores que es treballaran durant el projecte. En aquest moment el preu per hora és de 0,11€.

El cost d'Internet es calcularà de manera proporcional a les hores treballades tenint en compte que es paguen 50€ mensuals i això correspon a 0,07€ per hora.

Material	Unitats	Vida útil (anys)	Dies	Preu per unitat (€)	Cost total / Amortització (€)
Ordinador MSI GE60	1	4	129	1.055	93
Microsoft Office	1	3	33	149	4,5
Windows 10	1	3	129	145	17
Quartus II	1	-	30	0	0
Modelsim	1	-	24	0	0
Google Drive	1	-	-	0	0
Ganttproject	1	-	-	0	0
Racó FIB	1	-	-	0	0
Transport	6	-	-	8	48
Electricitat	1	-	465	0,11	51,15
Internet	1	-	465	0,07	32,55
TOTAL CG	16				246,2

Taula 20. Costos imputats genèricament.

5.1.3 Costos de contingència

En la *Taula 21* es mostren els costos de contingència. S'indica el cost sumant els costos de personal i els costos genèric, el percentatge de contingència, el cost de contingència un cop aplicat el percentatge i el pressupost sumant aquestes tres quantitats.

Tipus de cost	CPA+CG	Percentatge	Preu (€)	CPA+CG+Contingencial
Contingència	1.5276,2	15	2.291,43	1.7567,63

Taula 21. Costos de contingència.

5.1.4 Costos dels imprevistos

En la *Taula 22* es mostren els costos dels imprevistos. S'ha estimat que el risc que es produeixin imprevistos en aquestes dues activitats és d'un 20%. Per a calcular el preu de cada imprevist s'ha aplicat aquest percentatge al cost de l'activitat.

Activitat	Cost (€)	Risc	Preu (€)
Estudis i implementacions	3.240	20%	648
Depuracions i simulacions	2.565	20%	513
TOTAL imprevistos			1161

Taula 22. Costos dels imprevistos.

5.1.5 Resum pressupost

En la *Taula 23* es mostra una taula amb el resum del pressupost del projecte.

Tipus de cost	Preu (€)
CPA	15.030
CG	246,2
Contingències	2.291,43
Imprevistos	1.161
TOTAL	18.728,63

Taula 23. Resum del pressupost.

5.2 Control de Gestió

Per tal de controlar el pressupost al llarg del projecte, al final de cada tasca descrita en el diagrama de Gantt es realitzarà un seguiment de les hores reals treballades i del cost real de la tasca.

Per tal de realitzar el seguiment es crearà un document Excel on es guardaran diversos càlculs sobre cada tasca. S'utilitzaran les següents formules:

Desviació del cost = $(CE - CR) * HR$

Desviació del consum = $(HE - HR) * CE$

Desviació total = $(CE - CR) * HR + (HE - HR) * CE$

HE: hores estimades, CE: cost estimat, HR: hores reals, CR: cost real

Comparant aquest càlculs amb les estimacions del pressupost realitzades per a cada tasca, es podrà controlar la desviació respecte el pressupost inicial. També permet veure en quines tasques s'ha produït la desviació i determinar si la desviació s'ha produït per una diferència de cost o de consum. D'aquesta manera es podrà aprendre dels errors comesos en el pressupost i es tindran en compte en projectes futurs.

Per a solucionar aquestes desviacions s'han definit dues partides en el pressupost, una de contingència que correspon al 15% de la suma dels costos genèrics i de personal i una d'imprevistos que correspon al 20% del cost de les tasques "*Estudis i implementacions*" i "*Depuracions i simulacions*".

6 Sostenibilitat i compromís social

En aquest apartat es fa una reflexió sobre la sostenibilitat del projecte. Per a fer aquesta reflexió es responen un conjunt de preguntes relacionades amb la dimensió econòmica, social i ambiental del projecte en les tres fases de desenvolupament: projecte posat en producció, vida útil i riscos.

6.1 Dimensió econòmica

- **Has quantificat el cost (recursos humans i materials) de la realització del projecte? Quines decisions has pres per a reduir el cost? Has quantificat aquest estalvi?**

Sí que s'ha estimat aquest cost, el cost dels recursos humans està estimat en la *Taula 19* i el cost dels recursos materials en la *Taula 20*.

Per a reduir el cost del projecte s'ha utilitzat programari lliure que es pot descarregar i utilitzar de forma completament gratuïta.

No s'ha quantificat aquest estalvi ja que s'hauria trobar algun software de pagament que permeti implementar i analitzar circuits per a fer una comparació.

- **S'ha ajustat el cost previst al cost final? Has justificat les diferències (llicons apreses)?**

Sí que s'ha ajustat. Hi ha hagut algun imprevist però aquests s'ha pogut solucionar amb la part del pressupost dedicada als imprevistos tal i com s'explica en l'apartat 3.6.

- **Com es resol actualment el problema que vols abordar? En què millorarà econòmicament la teva solució respecte les existents?**

En aquest projecte es determina quina de les implementacions estudiades és la més eficient energèticament. De manera indirecta pot beneficiar econòmicament si algun fabricant opta per utilitzar aquesta implementació ja que al reduir el consum energètic es produeix un benefici econòmic.

- **Quin cost estimes que tindrà el projecte durant la seva vida útil? Es podria reduir aquest cost per a fer-lo més viable?**

El projecte no té associat un cost a la seva vida útil ja que es tracta d'un estudi acadèmic.

- **S'ha tingut en compte el cost dels ajustaments/actualitzacions/reparacions durant la vida útil del projecte?**

No s'han tingut en compte els costos associats als ajustaments/actualitzacions/reparacions durant la vida útil ja que es tracta d'un estudi acadèmic.

- **Poden produir-se escenaris que perjudiquin la viabilitat del projecte?**

Si es troba una implementació amb un cost energètic inferior i un error inferior no caldria utilitzar cap de les implementacions proposades en aquest estudi, però l'estudi d'aquestes implementacions s'hauria d'haver realitzat igualment per a fer la comparació.

6.2 Dimensió social

- **Què creus que t'aportarà la realització del projecte a nivell personal?**

La realització d'aquest projecte m'aportarà més coneixements sobre els disseny, implementació, simulació i avaluació de circuits ja que s'estudiaran circuits complexos que no s'han vist durant la carrera.

- **La realització del projecte ha implicat reflexions significatives a nivell personal, professional o ètic de les persones que han intervingut?**

Sí, a nivell personal ja que ha estat el projecte amb més feina que he realitzat fins el moment.

- **Com es resol actualment el problema que vols abordar? En que millorarà socialment (qualitat de vida) la teva solució respecte les existents?**

No aplica en aquest projecte.

- **Existeix una necessitat real del projecte?**

Sí, en l'apartat "1.3 Justificació" es compara el projecte a desenvolupar amb projectes previs i es justifica la necessitat real de realitzar-lo.

És justifica la necessitat real del projecte degut a que no s'ha trobat un estudi previ que compari amb profunditat els multiplicadors aproximats que s'estudiaran en aquest projecte.

- **Qui es beneficiarà de l'ús del projecte? Hi ha algun col·lectiu que es pugui veure perjudicat per projecte? En quina mesura?**

Tal i com s'ha explicat en l'apartat 1.3.2, els beneficiaris d'aquest estudi poden ser tant els fabricants del sector com els investigadors que puguin tenir en compte els resultats obtinguts.

- **En quina mesura soluciona el projecte el problema plantejat inicialment?**

Amb el temps establert per a realitzar aquest projecte és inviable analitzar totes les diferents implementacions de multiplicadors aproximats i analitzar l'efecte de combinar diferents implementacions entre elles. És per això que en aquest estudi només s'ha estudiat un petit conjunt de possibilitat i s'ha establert dins d'aquest conjunt quins són els millors dissenys i s'han obtingut resultats molt beneficiosos.

- **Poden produir-se escenaris que fessin que el projecte fos perjudicial per algun segment particular de la població?**
Al tractar-se d'un estudi, no té pensat un usuari final que usi el projecte, per tant, no pot ser perjudicial per a cap sector.
- **Podria crear el projecte algun tipus de dependència que deixes als usuaris en posició de debilitat?**
Al tractar-se d'un estudi no pot crear cap dependència als usuaris.

6.3 Dimensió ambiental

- **Has estimat l'impacte ambiental que tindrà el projecte? T'has plantejat minimitzar l'impacte, per exemple, reutilitzant recursos? Has quantificat aquesta reducció?**
L'impacte ambiental d'aquest projecte consisteix en els recursos materials que s'utilitzaran en el seu desenvolupament i estan quantificats en la *Taula 20*.
Per minimitzar l'impacte s'ha utilitzat software gratuït tal i com es pot observar també en la *Taula 20*.
- **Si tornessis a fer el projecte, podries realitzar-lo amb menys recursos?**
Sí, sobretot es podrien reduir el recursos humans ja que amb l'experiència adquirida al realitzar el projecte, podria realitzar un projecte similar amb moltes menys hores de treball. Per altre banda, no crec que es puguin reduir més els recursos material necessaris.
- **Com es resol actualment el problema que vols abordar? En què millorarà ambientalment la teva solució respecte les existents?**
En aquest projecte es determina quina de les implementacions estudiades és la més eficient energèticament. De manera indirecta pot afectar positivament al medi ambient si algun fabricant opta per utilitzar la implementació més eficient energèticament ja que es produiria un estalvi energètic.
- **Quins recursos estimes que es faran servir durant la vida útil del projecte? Quin serà l'impacte ambiental d'aquests recursos?**
La pregunta no aplica a aquest projecte al tractar-se d'un estudi.

- **El projecte permetrà reduir l'ús d'altres recursos? Globalment, l'ús del projecte millorarà o empitjorarà l'empremta ecològica?**

Sí, permetria reduir el consum elèctric si s'utilitzés alguna de les implementacions de multiplicadors presentades en l'estudi.

- **Poden produir-se escenaris que fessin augmentar l'empremta ecològica del projecte?**

La pregunta no aplica a aquest projecte al tractar-se d'un estudi.

7 Conclusions

En aquest projecte hem avaluat diferents implementacions de multiplicadors aproximats, hem estudiat l'efecte de combinar diferents implementacions en un mateix multiplicador i també hem estudiat l'efecte al ampliar el nombre de bits del multiplicador..

Els resultats obtinguts han estat molt bons ja que amb un error relatiu mitjà de només un 1% es pot aconseguir una reducció de consum de fins al 70%.

Hem pogut veure que per a diferents nombres de bits del multiplicador, diferents implementacions han resultat ser les més òptimes. Per a multiplicadors de 8 bits els més òptims són els que estan basats en multiplicadors AM2 i AM3 ja que tot i que aconseguixen una reducció de consum inferior als AM1, aquests últims tenen un error relatiu mitjà molt superior, de l'ordre del 20%.

Aquest fet canvia a l'ampliar el nombre de bits del multiplicador ja que l'error relatiu mitjà dels multiplicadors basats en AM1 es redueix significativament fins a aconseguir uns errors de ~1-3% tot i mantenir una reducció de consum del ~70%.

Cal tenir en compte altra vegada, que amb el temps disponible per a la realització del treball no és viable analitzar totes les possibles combinacions de multiplicadors aproximats, però s'ha intentat analitzar les millors versions en cada cas per tal d'establir una bona comparació. Aprofitant aquest punt, com a treball futur existeix la possibilitat d'ampliar el projecte amb l'estudi d'altres combinacions que no s'han estudiat en aquest projecte o d'estudiar altres implementacions de multiplicadors aproximats, ja que n'existeixen moltes més. I també de realitzar l'estudi de multiplicadors de 64 bits.

Cal remarcar, la utilitat que pot tenir el projecte per a investigadors d'aquest àmbit ja que poden fer servir els resultats obtinguts en aquest, per a establir comparacions amb altres implementacions de multiplicadors aproximats.

Per acabar, el projecte m'ha servit a nivell personal per aprofundir molt més en el disseny, implementació i avaluació de circuits, ja que s'han estudiat noves implementacions, s'ha fet un anàlisi del consum i de l'àrea dels circuits, que no s'havia treballat durant la carrera.

8 Referències

- [1] J. Han and M. Orshansky, "Approximate computing: An emerging paradigm for energy-efficient design," *2013 18th IEEE European Test Symposium (ETS)*, Avignon, 2013, pp. 1-6.
- [2] A. Pandey, M. R. Karri, P. Yadav, N. K. Y. B. and V. M.H., "Design and Analysis of Approximate Multipliers for Error-Tolerant Applications," *2018 IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly iNiS)*, Hyderabad, India, 2018, pp. 94-97.
- [3] P. Kulkarni, P. Gupta and M. Ercegovac, "Trading Accuracy for Power with an Underdesigned Multiplier Architecture," *2011 24th International Conference on VLSI Design*, Chennai, 2011, pp. 346-351.
- [4] Khaing Yin Kyaw, Wang Ling Goh and Kiat Seng Yeo, "Low-power high-speed multiplier for error-tolerant application," *2010 IEEE International Conference of Electron Devices and Solid-State Circuits (EDSSC)*, Hong Kong, 2010, pp. 1-4.
- [5] K. M. Reddy, Y. B. N. Kumar, D. Sharma and M. H. Vasantha, "Low power, high speed error tolerant multiplier using approximate adders," *2015 19th International Symposium on VLSI Design and Test*, Ahmedabad, 2015, pp. 1-6.
- [6] Masadeh, M., Hasan, O. and Tahar, S. (2018). *Comparative Study of Approximate Multipliers*. Available at: <https://arxiv.org/pdf/1803.06587.pdf> [Accessed 24 Sep. 2019].
- [7] Ramasamy, J. and Nagarajan, S. (2017). *Approximate Multiplier Techniques - A Survey*. Available at: <https://ijrest.net/downloads/volume-4/issue-5/pid-ijrest-45201710.pdf> [Accessed 24 Sep. 2019].
- [8] Quartus II: <http://fpgasoftware.intel.com/13.0sp1>
- [9] Modelsim: https://www.mentor.com/company/higher_ed/modelsim-student-edition
- [10] Google Drive: <https://drive.google.com>
- [11] Ganttter: <https://www.ganttter.com>

[12] Intel Quartus Prime Standard Edition User Guide: Power Analysis and Optimization:
<https://www.intel.com/content/www/us/en/programmable/documentation/xhv1529966780595.html>

[13] A. Sultan, A. H. Hassan and H. Mostafa, "A Compact Low-Power Mitchell-Based Error Tolerant Multiplier," *2018 New Generation of CAS (NGCAS)*, Valletta, 2018, pp. 130-133.

[14] P. Kulkarni, P. Gupta and M. Ercegovic, "Trading Accuracy for Power with an Underdesigned Multiplier Architecture," *2011 24th International Conference on VLSI Design*, Chennai, 2011, pp. 346-351.

[15] P. Yadav, A. Pandey, M. R. K., R. P. K.J., V. M.H. and N. K. Y.B., "Low Power Approximate Multipliers With Truncated Carry Propagation for LSBs," *2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS)*, Windsor, ON, Canada, 2018, pp. 500-503.

[16] PayScale <https://www.payscale.com/research/ES/Country=Spain/Salary>