



Departament d'Arquitectura
de Computadors

UNIVERSITAT POLITÈCNICA DE CATALUNYA



DISSENY D'UN BLOC ARITMÈTIC APROXIMAT PER A COMA FLOTANT

Treball de Fi de Grau

Autor: Jose Luis Estragués Muñoz

Directors: Enric Morancho i Ramon Canal

Departament d'Arquitectura de Computadors

Grau en Enginyeria Informàtica - Enginyeria de Computadors

Facultat d'Informàtica de Barcelona

Universitat Politècnica de Catalunya (UPC) – BarcelonaTech

26 de Gener de 2021

Resum

La majoria de processadors actuals utilitzen l'estàndard IEEE-754 per a nombres de coma flotant. Aquest estàndard presenta diverses fonts d'imprecisió en les operacions aritmètiques. També apareixen cada dia aplicacions com les Big Data i sistemes encastats, les quals no treballen utilitzant la mateixa precisió pel que algunes poden acceptar diferents graus d'error en els càlculs. Aquests fets justifiquen una nova rama d'optimització d'arquitectures hardware explota la tolerància a errors de càlcul per dissenyar processadors o acceleradors que operen amb inexactitud provocada intencionadament per reduir consum i temps d'execució.

En aquest projecte s'han dissenyat 63 circuits que aproximen el producte de nombres en coma flotant segons l'estàndard IEEE-754. Diverses tècniques d'aproximació s'han estudiat en aquest treball i s'ha creat un entorn de treball a mesura.

Resumen

La mayoría de procesadores actuales utilizan el estándar IEEE-754 sobre los números de coma flotante. Este estándar presenta varias fuentes de imprecisión en las operaciones aritméticas. También aparecen cada vez más aplicaciones como las Big Data, las cuales no trabajan utilizando la misma precisión por lo que algunas podrían aceptar diferentes grados de error en los cálculos. Esto justifica la nueva rama de optimización de arquitecturas hardware explota la tolerancia a errores de cálculo para diseñar procesadores y aceleradores que operen con inexactitud provocada intencionalmente para reducir consumo y tiempo de ejecución.

En este proyecto se han diseñado 63 circuitos que aproximan el producto de números en coma flotante según el estándar IEEE-754. Diversas técnicas de aproximación se han estudiado en este trabajo y se ha creado un entorno de proyecto a medida.

Abstract

The most part of actual processors use the standard IEEE-754 on floating point numbers. The standard has different sources of error at arithmetic operations. New applications such as Big Data and embedded systems use different ranges of precision. For this reason, some of them can support different error grades. This fact justifies a new branch of hardware design optimization that takes advantage of error tolerance to design processors or accelerators that operate with intentioned imprecision to save energy and execution time.

During this project, 63 circuits have been designed to approximate the product of floating points numbers following the standard IEEE-754. Multiple approximation techniques have been studied and a work environment has been designed specially for this project.

Índex

1	Context i abast	11
1.1	Context	11
1.1.1	Introducció	11
1.1.2	IEEE-754	12
1.1.3	Error en coma flotant	13
1.1.4	Actors	15
1.1.5	Projecte Drac	16
1.2	Justificació	17
1.3	Abast	18
1.3.1	Objectius	18
1.4	Restriccions	18
1.5	Metodologia	19
1.5.1	Etapas	19
1.5.2	Eines	19
2	Planificació del projecte	21
2.1	Definició de tasques	21
2.1.1	Gestió de projectes	21
2.1.2	Investigació	22
2.1.3	Part experimental	22
2.1.4	Anàlisi	22
2.1.5	Documentació del projecte	22
2.1.6	Reunions de seguiment	22
2.1.7	Defensa del treball	23
2.2	Recursos	23
2.2.1	Recursos humans	23
2.2.2	Recursos materials	23
2.3	Gestió de riscos: plans alternatius	23
2.4	Desviacions	24
3	Pressupost i sostenibilitat	27
3.1	Pressupost	27
3.1.1	Agent	27
3.1.2	Material	27
3.1.3	Cost total	28
3.1.4	Control de la gestió	28
3.1.5	Desviació del pressupost	29
3.2	Sostenibilitat	30
3.2.1	Dimensió econòmica	30

3.3	Dimensió ambiental	30
3.4	Dimensió social	31
4	Verificació i tests	33
4.1	Verificació de components	33
4.2	Eines de Quartus	35
4.2.1	Timing analyzer	35
4.2.2	PowerPlay Power Analyzer	36
4.3	Sistema d'automatització de tasques	37
4.3.1	Generació d'operands	37
4.3.2	Testbench	38
4.4	Comparació de resultats	38
5	Model Base	41
5.1	Característiques del circuit	41
5.2	Mòdul Signe	42
5.3	Mòdul exponent	42
5.4	Wallace Tree de 24 bits	43
5.5	Mòdul normalitzador	44
5.6	Multiplicador Coma Flotant 32 bits	46
6	Models aproximats	49
6.1	Aproximacions mòdul Wallace Tree	49
6.2	Models Segmentats	50
6.2.1	Sumadors segmentats	50
6.3	Models especulatiu	51
6.3.1	Sumadors especulatiu	51
6.3.2	Sumadors especulatiu amb compensador	53
6.4	Sumador Low OR	54
6.5	Full Adders Aproximats	55
6.6	Aproximacions en el mòdul Exponent	58
6.7	Aproximacions en el mòdul Normalitzador	58
7	Anàlisi	59
7.1	Resultats aproximació Wallace Tree	59
7.1.1	Aproximació en la suma de productes parcials	60
7.1.2	Aproximació en la generació de productes parcials	65
7.1.3	Approximate Adders Wallace Tree	70
7.1.4	Estudi entre tipus	73
7.2	Aproximacions de l'exponent	74
7.3	Aproximacions mòdul Normalitzador	75
7.4	Conclusions de l'anàlisi	76
8	Conclusions	79

Índex de figures

1.1	Gràfica llei de Moore.	12
2.1	Diagrama de Gantt.	25
2.2	Diagrama de Gantt amb desviacions.	26
4.1	Add Files.	34
4.2	Board Settings.	35
4.3	Simulació ModelSim.	36
4.4	Etapes Power Analyzer.	36
4.5	Test Bench.	39
4.6	Procés de Verificació i Tests.	40
5.1	Mòdul Exponent.	43
5.2	Mòdul Wallace Tree.	44
5.3	Mòdul Normalitzador.	45
5.4	Diagrama del Multiplicador de Coma Flotant de 32 bits.	47
6.1	Mòdul Especulador.	52
6.2	Mòdul Compensador.	54
6.3	Sumador Especulatiu amb Compensadors.	55
6.4	Sumador Low OR	56
6.5	Taula de veritat AA[1...5]	57
6.6	Taula de veritat AA[6...12]	57
7.1	Boxplot Models Low OR	61
7.2	Boxplot Models Segmentats	62
7.3	Boxplot Models Especulatius	63
7.4	Boxplot Models Compensador	64
7.5	Boxplot Models AA	66
7.6	Boxplot Models Wallace Tree Low OR.	67
7.7	Boxplot Models Wallace Tree Segmentat	68
7.8	Boxplot Models Wallace Tree Especulatiu.	69
7.9	Boxplot Models Wallace Tree Compensador	70
7.10	Boxplot Models Wallace Tree AA 25%	72
7.11	Boxplot Models Wallace Tree AA 50%	73
7.12	Optimitzacions de models aproximats.	77

Índex de taules

1.1	Distribució dels bits i el seu significat en el IEEE-754	13
1.2	Tipologia i valors del IEEE-754 de 32 bits.	13
3.1	Cost equips informàtics.	28
3.2	Recopilació de despeses.	28
3.3	Desviacions.	29
6.1	Taula de veritat del mòdul Especulador.	53
6.2	Lògica Approximate Adders.	56
7.1	Mètriques models Low OR.	60
7.2	Mètriques models Segmentats.	61
7.3	Mètriques models Especulatiu.	63
7.4	Mètriques models Compensador.	64
7.5	Mètriques models Approximate Adders.	65
7.6	Mètriques models Low OR Wallace Tree.	66
7.7	Mètriques models Segmentats Wallace Tree.	67
7.8	Mètriques models Especulatiu Wallace Tree.	68
7.9	Mètriques models Compensador Wallace Tree.	69
7.10	Mètriques models AA 25% Wallace Tree.	71
7.11	Mètriques models AA 50% Wallace Tree.	72
7.12	Mètriques models Exponent.	75
7.13	Mètriques model Normalització.	75

Capítol 1

Context i abast

Aquest és un treball de fi de grau (TFG) del Grau en Enginyeria Informàtica, especialització en Enginyeria de Computadors, fet a la Facultat d'Informàtica de Barcelona de la Universitat Politècnica de Catalunya dirigit pels professors Enric Morancho i Ramon Canal.

1.1 Context

1.1.1 Introducció

Les operacions de coma flotant històricament han estat implementades en un mòdul apart de la Central Processor Unit (CPU) [19]. Els equips informàtics que no disposen d'aquesta unitat funcional, realitzen les operacions per software, a través de biblioteques especialitzades. Actualment, la reducció de la mida dels transistors ha facilitat la integració d'aquesta unitat funcional dins de la CPU. Això permet una reducció considerable en el temps de resposta d'una instrucció de coma flotant (latency) i també un increment considerable de la quantitat d'operacions que es poden realitzar per unitat de temps (bandwidth). Aquest fet es descriu en la Llei de Moore [15] on s'observa que cada dos anys la mida dels transistors es redueix a la meitat.

La tendència de mercat ha estat oferir processadors cada cop més eficients en termes d'energia i capacitat de còmput. Però tal com es mostra en la figura 1.1, la corba s'ha aplanat. Per aquesta raó, s'han de buscar tècniques per solucionar nous reptes que apareixen: la necessitat d'estalviar energia o proporcionar un rendiment suficient pel correcte funcionament de la gran estructura de l'Internet of Things entre altres.

Per altra banda, un altre factor que cada dia guanya més importància és el fet que certes aplicacions són tolerants a imprecisions amb les dades que treballen o els resultats que produeixen. Aquesta és una de les principals motivacions del projecte i serà un factor que influirà a l'estudi considerablement.

En aquest treball, s'estudiaran diverses tècniques d'aproximació aplicables al disseny hardware d'una unitat de multiplicació de coma flotant. La unitat operarà amb operands que segueixen el estàndard IEEE-754. Tot i que no es treballa directament amb dissenys de Floating Point Unit (FPU) comercials, seguirem l'estàndard ja que ens facilita que el treball sigui aplicable en altres dissenys d'FPUs.

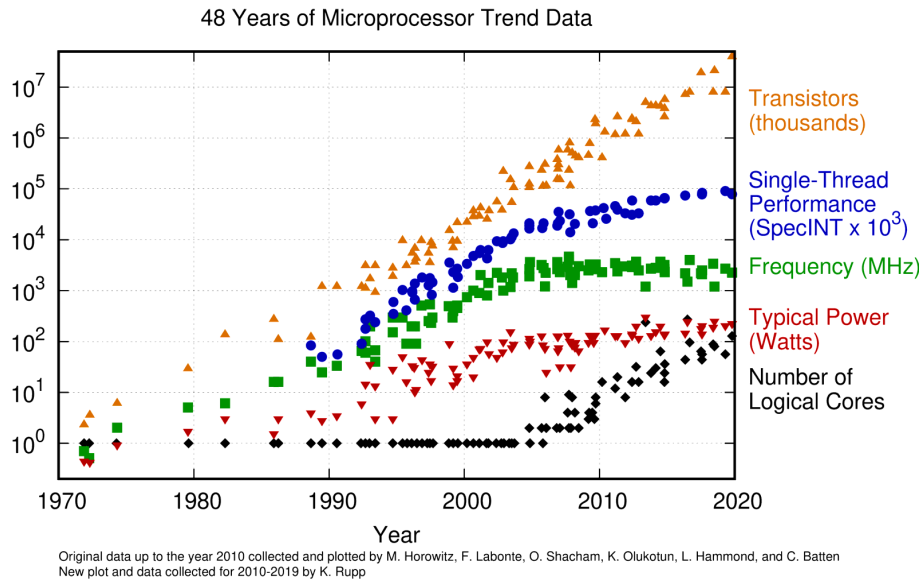


Figura 1.1: Gràfica llei de Moore.

Representació de la quantitat de **transistors**, **freqüència**, **transistors en milers**, **rendiment per thread**, **consum elèctric** i el **número de cores** [16].

1.1.2 IEEE-754

El Institute of Electrical and Electronic Engineers (IEEE) va estandarditzar el format de coma flotant en el 1985 [12]. Va establir un referent en la representació dels nombres reals en binari, també abordant molts problemes coneguts en les diverses implementacions que coexistien en aquella època. El document defineix tal com s'esmenta en[13]:

- **Formats aritmètics:** conjunt de dades de coma flotant binaris i decimals, que consisteixen en la següent tipologia de números:
 - Zero amb signe.
 - Números desnormalitzats, aquell que no es poden normalitzar degut a que tenen un exponent massa petit.
 - Infinit amb signe.
 - Valors especials "no numerics", Not a Number (NaN).
- **Formats d'intercanvi:** ajuda a definir cadena de bits que s'utilitzen per intercanvi de dades.
- Propietats que ha de satisfer dels **números arrodonits**.
- Un conjunt **d'operacions aritmètiques**.
- **Excepcions** pròpies tal com la divisió entre 0.

Entre aquestes totes característiques esmentades també trobem la estandarització de aquest nombre de coma flotant en binari. Aquesta representació és necessària pel fet que les màquines actuals treballen amb registres de capacitat potència de 2 bits. Els formats més utilitzats són els de 32, 64 i 128 bits, però en aquest treball també estudiarem el format de 32 bits que a dia d'avui és l'arquitectura més adequada per petits sistemes encastats. En la Taula 1 es mostra els bits utilitzats per a cada representació i el seu significat.

Tamany	Signe	Exponent	Mantissa
16 bits	bit [15]	bit [14...10]	bit [9...0]
32 bits	bit [31]	bit [30...23]	bit [22...0]
64 bits	bit [63]	bit [62...52]	bit [51...0]
128 bits	bit [127]	bit [126...112]	bit [111...0]

Taula 1.1: Distribució dels bits i el seu significat en el IEEE-754 .

Els nostres dissenys i optimitzacions poden ser aplicables a estructures que segueixin el mateix estàndard.

Aquest projecte es centrarà en el números normalitzats o normals. Segons els valor de exponent i mantissa, es pot identificar el tipus de nombre. En la següent taula es presenta dits valors crítics per una arquitectura de 32 bits.

	Signe	Exponent	Mantissa
Zero positiu	0	0	0
Zero negatiu	1	0	0
Infinit positiu	0	255	0
Infinit negatiu	1	255	0
Not a Number	-	255	$\neq 0$

Taula 1.2: Tipologia i valors del IEEE-754 de 32 bits.

La resta de valors es poden definir com números normals. Aquests tenen la característica que s'emmagatzemen en notació científica. El rang de aquests números és de $-3,4 \cdot 10^{38}$ fins $3,4 \cdot 10^{38}$.

1.1.3 Error en coma flotant

Durant la realització del projecte es parla reiteradament de l'error causat pels models aproximats en operacions. Aquest fet no implica que els estàndards, en aquest cas el IEEE-754, calculin sense errors. Un fet és clau, la precisió dels nombres reals és infinita, en canvi les combinacions de números que es poden fer en registres dels ordinadors actuals són limitades. Per aquesta raó cal fer menció de les possibles fonts d'errors contra qui lluiten les actuals unitats funcionals de coma flotant. En aquesta secció es fa un recorregut per aquestes principals causes d'errors.

Error en l'arrodoniment

A diferència del càlcul amb nombres enters, el IEEE-754 introdueix l'ús d'arrodoniments en les seves operacions. Tres causes justifiquen el seu ús [10]:

- **Denominadors massa grans:** hi ha una relació proporcional entre el nombre de bits necessaris per representar un nombre decimal i el mida del denominador de la fracció irreductible. Durant més gran el denominador, més bits es necessiten. Per exemple, es necessiten més de 3 bits per representar els decimals de $1/1000$.
- **Números periòdics:** els números amb decimals periòdics tenen un patró de decimals que es repeteix infinitament. Per exemple, $1/3$.

- **Números irracionals:** aquests números disposen de decimals infinits que no segueixen cap patró. Per exemple, el número pi.

Mètodes d'arrodoniment

En aquest subapartat s'explica els mètodes d'arrodoniment definits per l'estàndard [17] que se segueix durant el treball.

- **Round to Nearest:** és el mètode per defecte. L'arrodoniment es fa cap el bit 0 més proper. Hi ha un 50% de possibilitats d'arrodonir cap a una direcció concreta.
- **Round toward 0:** l'arrodoniment es fa cap el número de menys valor absolut. És a dir, cap al 0 absolut. La direcció de l'arrodoniment depèn exclusivament del signe del valor.
- **Round toward +infinity:** l'arrodoniment es fa cap el infinit positiu.
- **Round toward -infinity:** l'arrodoniment es fa cap el infinit negatiu.

Comparacions

Els problemes de precisió que hem descrit anteriorment poden causar que dos números iguals però calculats de forma distinta, no siguin iguals en una comparació. Aquest fet és important a tenir en compte per a programadors. La bona pràctica que es recomana [4] en el següent cas és calcular la proporció d'error en comptes de l'error absolut. A continuació tenim un pseudocodi dels dos casos.

Incorrecte: `if (abs(a - b) < error)`

Correcte: `if (abs((a - b) \ b) < error)`

En el primer cas utilitzem una mètrica d'error coneguda com a èpsilon i que és útil per a quan necessitem la mateixa precisió en qualsevol rang de valors, tant petits com grans. En canvi, el segon cas utilitzem la proporció d'error. Aquesta mètrica es manté constant per a tot rangs de valors i és útil per identificar els casos a on els errors són induïts pel sistema de coma flotant.

Aquest últim mètode presenta una sèrie de casos límits a tenir en compte[4]:

- Els dos operands són 0.0: el resultat és NaN i provoca una excepció a part de la comparació que sempre és falsa.
- Operand b és 0.0: la divisió dóna resultat infinit i per tant sempre es falsa la comparació.
- Dos operands propers al zero: sempre retorna fals.

Propagació de l'error

Un altre factor a tenir en compte és la propagació de l'error. Aquests errors que s'introdueixen en els càlculs de forma inconscient poden donar peu a errors més grans. Les operacions de multiplicació i divisió són segures pel fet que qualsevol combinació de magnituds es veuen reflectides en el resultat. El cas contrari és la suma i la resta. Són operacions que si els operands són de magnituds molt diferents, és possible que no hi hagin efectes en el resultat.

En els dos casos, l'error es pot propagar i en conseqüència augmentar si es combina amb altres operacions similars. Un clar exemple són els algorismes iteratius.

1.1.4 Actors

En aquesta secció farem un petit anàlisi dels actors que es veuran afectats per aquest treball d'una forma directa i indirecta. També s'ha de tindre en compte que encara que és un treball de recerca i que els resultats no són extrets de casos reals sinó d'un entorn de laboratori.

Els actors que es veuen involucrats directament són:

- **Investigador:** en aquest cas coincideix amb l'autor del treball. És l'encarregat de fer la recerca, aplicar els coneixements en la part pràctica, extreure resultats, analitzar-los i redactar el manuscrit amb tot el que allò implica.
- **Directors:** l'Enric Morancho i el Ramon Canal són els tutors d'aquest treball. El seu rol és donar suport als possibles dubtes o problemes que apareguin durant la realització del treball a part d'orientar a l'estudiant.

Per una altra banda, els actors que no intervenen en la realització del treball són:

- **Comunitat científica:** el treball està disponible per tota la comunitat científica interessada en el desenvolupament d'aquest tema. Els resultats i conclusions són lliures de ser utilitzats i difosos sempre respectant el crèdit dels autors tal com aquest treball respecta els autors de les fonts bibliogràfiques.
- **Fabricants de hardware:** una gran quantitat de dispositius electrònics incorporen una petita unitat de processament. Les millores proposades es poden aplicar en tots aquests en els quals els seus desenvolupadors trobin adient.
- **Usuari:** tot aquell dispositiu que utilitzi un disseny aproximat proposat per aquest treball, serà utilitzat per usuaris o bé per altres màquines que acaben oferint un servei a usuaris. En tot cas, tota mena de persones són el possible consumidor final donat a la gran varietat d'ús que es pot donar al treball.

Aquest són els actors que considerem en el treball. Totes les decisions que influeixin en el desenvolupament del mateix serà considerant èticament els interessos i necessitats dels actors.

1.1.5 Projecte Drac

Citant textualment la font [6] el projecte DRAC té com a missió:

L'associació en tecnologies emergents DRAC (Designing RISC-V-based Accelerators for next generation Computers), dissenyarà, verificarà, implementarà i fabricarà un processador de propòsit general d'alt rendiment que incorporarà diferents acceleradors basats en la tecnologia RISC-V, amb aplicacions específiques en el camp de la seguretat, la genòmica i la navegació autònoma. RISC-V és un conjunt d'instruccions d'arquitectura oberta definida per la Fundació RISC-V. Aquesta fundació ha atret més de 100 institucions internacionals, incloses universitats, centres de recerca i empreses de tot el món. RISC-V és modular i extensible a la qual se li poden incorporar múltiples acceleradors al disseny.

DRAC desenvoluparà un processador RISC-V de propòsit general fora de sèrie que superarà als actuals i futurs en termes de processadors de disseny. A més, DRAC dissenyarà el suport de maquinari requerit per accelerar tres dominis d'aplicacions emergents: seguretat, medicina de precisió i navegació autònoma.

- Dissenyar un processador segur és un desafiament, especialment si considerem l'arribada futura d'ordinadors quàntics que faran vulnerables els esquemes criptogràfics implementats en els processadors actuals. Les càrregues de treball de la medicina de precisió prevaldran en el futur.
- L'anàlisi de medicina de precisió a nivell de població requereix una gran potència de càlcul. Ser capaç de realitzar aquests càlculs de manera més ràpida i eficient permetrà que la medicina de precisió es converteixi en una realitat en un futur proper.
- Finalment, amb l'arribada dels cotxes autònoms, els cotxes estaran equipats amb processadors més potents. Reduir la potència i el cost d'aquests processadors és fonamental perquè les actuacions autònoms arribin a el mercat en un futur proper.

Les aproximacions estudiades en aquest treball es presentaran com a proposta pel processador que està en desenvolupament, on podríem veure el resultat del treball aplicat a un processador real o a un accelerador que treballi conjuntament.

1.2 Justificació

En aquesta secció es presenten les raons primàries que justifiquen la necessitat de realitzar aquest treball. Com es pot intuir en les diferents seccions anteriors, hi ha múltiples raons per la qual una unitat de coma flotant aproximada és d'utilitat en diferents casos. Certes aplicacions toleren un grau d'error superior al que de forma natural introdueix l'estàndard IEEE-754. Aquest fet s'aprofita per dissenyar unitats de càlcul aproximades que redueixin el consum i temps respecte a les unitats no aproximades.

Aquestes dos necessitats tenen arrel en un paradigma de l'arquitectura de computadors on veiem que difereix de la tendència que ha seguit els últims anys. Tal com s'ha introduït en el context, podem veure que la Llei de Moore donava una observació que és complia durant una gran quantitat d'anys. El fet que el volum dels transistors es redueix ajudava a incrementar els elements que podien entrar dins del xip de la CPU. La proximitat dels elements feia que cada cop els processadors fossin més eficients i incorporassin més utilitats. Inclús la integració de la memòria caché dins del xip és un resultat d'aquesta observació. Cada dos anys teníem processadors més eficients i ràpids.

El punt d'inflexió es troba en el fet que aquestes tècniques ja no són efectives per incrementar el rendiment de forma profitosa. D'aquesta manera, s'intenta explotar el rendiment d'amb altres mètodes tal com paral·lelisme (multicore, multithreading, clustering, entre altres) i reinventant la forma en la qual ja s'usaven els recursos existents (segmentació, protocols de caché o fins i tot predictors d'instruccions o accessos a memòria). Totes aquestes tècniques tenen en comú que no tenen com a límit no alterar ni el resultat de les operacions.

La proposta del treball trenca amb aquesta norma i permet introduir una quantitat d'error en el resultat amb objectiu guanyar velocitat de còmput. La inexactitud del resultat està justificat per la naturalesa de les aplicacions a que va dirigida. Des de sistemes de temps real que necessiten ràpida resposta del dispositiu, aquestes màquines normalment treballen com a controladors i/o llegint sensors. Elements que en la seva aplicació pràctica accepten certa falta de precisió. Per aquestes raons, hi ha una alta possibilitat que aproximar circuits hardware sigui una nova disciplina en el món de l'arquitectura de computadors.

1.3 Abast

Donat el ampli abast del tema escollit i el reduït temps de treball que ens faciliten, es mandatori acotar la profunditat i els temes que es tracten en el projecte. Les instruccions de coma flotant són un tema ben conegut dins del món de l'arquitectura de computadors i del disseny hardware. Per això, tindre uns objectius clars i concisos son la clau per un treball eficaç. Objectius indefinits o massa ambiciosos poden portar nos per camins laberíntics o que mai arriben a un destí. En aquesta secció es defineixen els objectius i l'abast del projecte.

1.3.1 Objectius

Pensant que el tema d'investigació és aproximar un multiplicador de coma flotant, és lògic pensar que el nostre objectiu és tindre un nou disseny hardware que estalviï en energia i temps de computació, tot això reduint en la mesura del possible l'error causat per l'aproximació. Però aquest no és el cas, perquè no valorarem obtenir mètriques espectaculars ni revolucionaries, sinó que ens centrarem en:

- **Anàlisi de disseny hardware:** detectar carències del circuit i establir criteris d'optimització.
- **Recopilar tècniques d'aproximació:** qualsevol mètode aplicable al treball serà contingut interessant a considerar.
- **Analitzar resultats:** raonar sobre les conseqüències de les modificacions i quins avantatges i inconvenients presenten.

1.4 Restriccions

Els números de coma flotant tenen un tractament complex a causa de factors com la seva representació en bits, arrodoniments, excepcions/interrupcions específiques... Segons el tipus d'operació aritmètica els operands han de complir certes restriccions que necessitaran preprocessar i/o post-processar els operands i resultats. En aquesta secció detallarem els aspectes que s'ha decidit no incloure en el treball per raons logístiques.

Números de coma flotant especials

L'estàndard IEEE-754 fa una classificació de diferent tipus de números de coma flotant. Es diferencien segons el seu valor i poden tindre una interpretació diferent segons tipus. En la secció 1.1.2 es fa menció d'aquesta classificació, en aquest treball només es dissenyen els mòduls i aproximacions pels números coneguts com a normals.

Excepcions

Tal com hem presentat en la secció del IEEE-754, les operacions de coma flotant tenen excepcions pròpies. La detecció d'aquestes excepcions implicaria afegir lògica a les nostres implementacions. Per aquesta raó, no considerarem aquests casos per centrar el treball en les aproximacions.

1.5 Metodologia

En la següent secció presentarem la metodologia escollida per realitzar aquest treball. S'ha dividit l'explicació entre les etapes del treball i les eines utilitzades per realitzar el mateix.

1.5.1 Etapes

Aquestes són les etapes en les que s'ha dividit el treball:

- **Establir i estudiar un referent de multiplicador de coma flotant:** les instruccions de coma flotant es poden implementar de múltiples formes. Fins i tot, s'ha definit uns diversos estàndards tant per part d'empreses privades com Microsoft o com del Institute of Electrical and Electronics Engineers . Per tant, no és un pas trivial saber de quin model serà el nostre referent pels models aproximats proposats.
- **Dissenyar un multiplicador de coma flotant base:** per realitzar les aproximacions es partirà d'un model base, el qual hem d'implementar nosaltres. Per raons òbvies, no podem partir d'un model comercial a causa de propietats intel·lectuals. S'ha de considerar diverses arquitectures del mateix model com ara 32, 64 bits..
- **Estudiar tècniques d'aproximació:** referent a aquest punt, es tracta de recollir mètodes d'aproximació per a circuits hardware. Es basa a entendre el funcionament de l'algoritme i/o flux de dades.
- **Adaptar les tècniques estudiades al nostre model base:** un cop hem escollit un seguit de possibles modificacions, les hem d'implementar en el nostre model base. Aquest procés s'ha de donar respectant al màxim l'estructura inicial. També incorporar de manera combinada diverses modificacions en el cas que siguin compatibles.
- **Proposar tècniques d'aproximació noves:** si la logística ho permet, es repetirà el procés anterior amb mètodes propis de l'autor.
- **Analitzar resultats:** extreure mètriques i analitzar els resultats és un objectiu en si de l'estudi d'un projecte d'aquest tipus. S'ha d'establir un entorn apte per aquest propòsit. Tot això ha d'anar acompanyat d'un raonament i conclusions crítiques.

1.5.2 Eines

Quartus II i Modelsim són les eines que mereixen menció degut a la seva rellevància durant el desenvolupament del treball.

Quartus II

Quartus II és un software dedicat al anàlisi i síntesis de Hardware Description Language. És una eina desenvolupada per Altera i la seva versió Prime és de llicència oberta. S'utilitzarà com entorn de programació i també per examinar diagrames Resistor-Transistor Logic (RTL).

Modelsim

És una eina també de llicència oberta en la versió Prime que ens ajudarà a fer la simulació del nostre codi VHDL i així debugar i establir la quantia d'error als resultats.

Capítol 2

Planificació del projecte

Aquesta secció detalla la planificació del projecte. La duració del treball és d'aproximadament 450 hores distribuïdes des del 1 de setembre de 2020 fins al 15 de gener de 2021. La càrrega de treball és d'aproximadament 4 hores per dia.

2.1 Definició de tasques

La realització del projecte es basa en diferents etapes compostes per tasques diverses. En aquest capítol es defineixen les tasques que hem plantejat i que hem situat cronològicament en la figura 2.1. En aquest treball s'ha donat una gran importància a la planificació pel fet que les diferents etapes depenen de l'anterior. Per aquesta raó, hem de dedicar el temps necessari i proporcional a cada una de les etapes per així poder fer que el projecte avanci. En el següent llistat tenim les diferents etapes i tasques de la mateixa definida.

2.1.1 Gestió de projectes

Aquesta és l'etapa inicial del projecte. Després de registrar el tema amb l'ajuda dels directors, es dedica una gran quantitat de temps a planificar i gestionar el projecte. S'han de marcar temes tan importants com la durada de les diferents tasques, la duració d'aquestes, l'abast del treball... Per això, s'utilitzarà l'ajuda de les indicacions de l'assignatura complementària a aquesta etapa Gestió de Projectes (GEP). Les tasques realitzades en aquesta etapa són:

- **Context i abast:** en les seccions anteriors ja hem definit quin és el context i abast del treball. Informar-se del context en el qual el treball es realitzarà és vital. Saber si ja hi ha aproximacions o les causes per les quals el treball és necessari i justificat. Acotar l'abast és clau per obtenir l'objectiu sense càrregues de treball excessives ni tampoc treball innecessari.
- **Planificació del projecte:** Definir les etapes i tasques. Organització temporal d'aquestes. També planificació dels recursos tant naturals com humans i els possibles obstacles que es poden preveure.
- **Pressupost i sostenibilitat:** Com en tots els projectes, el pressupost un factor molt cop limitant i que juga una gran importància en el èxit d'aquest. Ser conscient de la sostenibilitat del treball és part de la nostra responsabilitat com a part de la societat, per això hem d'estudiar les possibles conseqüències pel medi ambient que pot desencadenar les accions preses.
- **Definició definitiva:** perfilar detall i correccions de les tasques anteriors.

2.1.2 Investigació

Aquesta etapa és una de les més llargues i alhora important del projecte, perquè el contingut es veurà fortament influenciat per la qualitat de la recerca. Es realitzarà durant la pròpia gestió del projecte perquè són dues etapes independents i també és important per la definició del context i saber quin abast és adient.

2.1.3 Part experimental

Aquesta etapa és la més crítica del projecte. Pel fet que és on més inconvenients es poden presentar. No acabar l'etapa suposaria no tindre resultats útils per analitzar. D'aquesta manera no es disposaria d'una discussió sobre diferents dissenys el qual una gran motivació del treball. Les tasques que realitzarem són:

- **Entorn de treball:** el disseny hardware és un àmbit pel qual un ordinador necessita preparació prèvia. En aquest temps es dedica a instal·lar software necessari i també molt important establir un disseny base en el qual se sortiran tots els dissenys posteriors. Així utilitzar un referent de mètriques per les possibles millores.
- **Implementació:** adaptar les millores que s'urgeixin de l'etapa d'investigació i també intentar crear millores pel model base.
- **Tests:** Un cop una implementació aquesta llesta, cal validar el disseny amb un test. Per això, s'ha de dissenyar una sèrie de proves fetes especialment per aquest entorn.

2.1.4 Anàlisis

Les mètriques del disseny s'han d'analitzar, buscar les raons que hi ha darrere dels números i fer conjectures de possibles millores futures. També s'han definit quins paràmetres s'obtidran les mètriques. Aquesta és una etapa que es va madurant durant tota la part pràctica i que per aquesta raó el temps assignat es dedica a la redacció d'aquest.

2.1.5 Documentació del projecte

Aquesta etapa es reserva per escriure la memòria del treball. Aquesta s'ha de presentar. Les tasques són:

- **Recollir dades:** compilar les mètriques obtingudes.
- **Material complementari:** el document es recolzarà en fotos, gràfiques, diagrames, per transmetre un missatge més clar i elegant. En aquesta tasca prepararem aquest material.
- **Redacció:** elaborar el codi en Latex i la redacció i correcció del text que inclourà la memòria del treball.

2.1.6 Reunions de seguiment

Les reunions es realitzen de forma telemàtica per la situació actual del virus COVID-19. Es faran de forma regular durant tota la duració del projecte. Durant la realització de les mateixes es comparteix els avenços fets per part de l'autor i els directors donen consells i s'acorden futurs detalls.

2.1.7 Defensa del treball

Un cop la memòria està completa, es pot començar a preparar la defensa. Per presentar pròpiament el treball, s'ha de preparar material visual com diapositives acompanyades pel material inclòs en la memòria. També és important escriure un guió i controlar factors tan importants com la duració de la defensa.

2.2 Recursos

Planejar els recursos serà útil per saber que material podem fer servir. En la definició de tasques ja hem fet servir un diagrama de Gantt per organitzar el temps, el nostre recurs més valuós. En aquesta secció s'acabarà de mencionar els recursos que es faran servir.

2.2.1 Recursos humans

Durant el treball tenim 4 persones involucrades en el projecte, l'autor i els dos directors s'han mencionat en la secció d'actors. També és important mencionar als professors de GEP que hi han transmet el coneixement necessari perquè el treball estigui millor organitzat i una memòria de més qualitat.

2.2.2 Recursos materials

En aquesta secció es fa una petita recopilació del material utilitzat durant el treball:

- **Ordinador DELL:** ordinador d'escriptori en el qual es busca informació, redactarà, etc. És un Intel I5 d'octava generació amb 8 GB de RAM i una gràfica de NVIDIA GeForce 1050. Aquest hardware opera un sistema operatiu Ubuntu i el seu punt fort són els dos grans monitors que disposa on el qual es poden treballar amb diversos documents alhora.
- **Ordinador Personal:** ordinador personal de l'autor el qual disposa de grans prestacions: Processador AMD RYZEN 5 1600, 32 GB de RAM i una targeta gràfica AMD Radeon RX 580. No és necessària el rendiment però aquest serà d'utilitat per reduir els llargs temps de compilació dels dissenys hardware. Tota la programació és realitzarà en aquest equip.
- Accés a **Internet**.
- **Software:** Sistemes operatius, editors de textos, Skype, Quartus II, Modelsim, navegadors... També sense mencionar els serveis del núvol com l'emmagatzematge de Google Drive.

2.3 Gestió de riscos: plans alternatius

En aquesta secció es defineixen els possibles riscos que es poden plantejar durant la realització del projecte. Aquests inconvenients poden fer que el projecte no s'acabi o no compleixi les expectatives esperades. Aquests són els riscos:

- **Deadline** (alt risc): la data límit d'entrega és la raó més important per la qual s'ha fet tota la planificació temporal del projecte. Si el treball realitzat no és correcte per la data d'entrega, s'hauria d'ampliar el termini, cosa que provocaria molèsties addicionals. Una alternativa si és pot preveure, és augmentar el temps de treball diari.
- **Dissenys hardware** (alt risc): realitzar dissenys hardware és una tasca complexa i que requereix molta dedicació i hores de debugar errors. Per aquesta raó, és una realitat que qualsevol complicació pot requerir hores i hores de treball. En el cas que un model o implementació no funcioni, s'estudiarà si és convenient continuar amb la millora concreta.

- **COVID 19** (risc mitjà): vivim una situació excepcional en aquest segle, el virus COVID 19 ha canviat la nostra forma de treballar, concebre les relacions socials i estil de vida. Per evitar riscos, en el projecte es prenen mesures com les reunions telemàtiques. El pitjor escenari és que l'autor es veïés afectat pel virus per una llarga temporada. En aquest cas, s'hauria de reorganitzar tota la planificació i adaptar-la pel temps perdut.

2.4 Desviacions

Durant la realització del projecte s'han produït desviacions en alguns paràmetres que es van estimar en un principi. Majoritàriament, els temps previstos no han acabat quadrant amb el necessari per acomplir el projecte. Certes etapes han requerit més temps.

Les fases d'implementació de l'entorn de treball i de les aproximacions s'han allargat donat al risc mencionat anteriorment sobre els dissenys hardware. Per tal de complir les Deadlines, s'hi ha hagut de modificar el diagrama de Gantt inicial amb les següents modificacions:

- **Entorn de treball:** s'han necessitat 8 dies (32 hores) extres per finalitzar el disseny de l'entorn de treball; model base i mètodes de testatge i verificació.
- **Implementació:** durant a les modificacions de l'etapa anterior, s'ha posposat l'inici de la implementació de les aproximacions. També s'han necessitat 16 dies (64 hores) extres.
- **Anàlisis:** s'ha modificat el període de treball però no la duració.
- **Conclusions:** 2 dies (8 hores) extres s'han requerit per realitzar les conclusions.
- **Documentació:** el període de dies ha incrementat. S'ha considerat començar aquesta etapa de documentació concurrentment amb etapes anteriors. Les hores de treball no s'han modificat sinó que s'han repartit entre més dies.
- **Reunions:** el període de reunions s'ha incrementat 11 dies per tal de preparar la defensa del treball. No s'han necessitat hores extres.

En total s'han necessitat un total de 104 hores que a part de l'impacte en la planificació temporal, també ha tingut un impacte en el pressupost que s'explicarà en la següent secció.

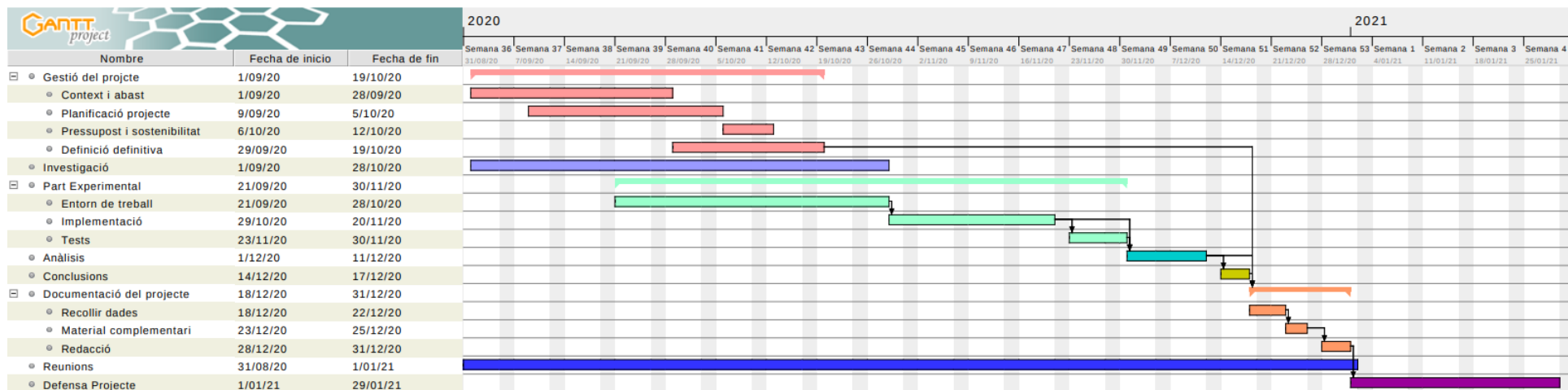


Figura 2.1: Diagrama de Gantt.
 Diagrama de Gantt de la realització del projecte. Elaboració pròpia.

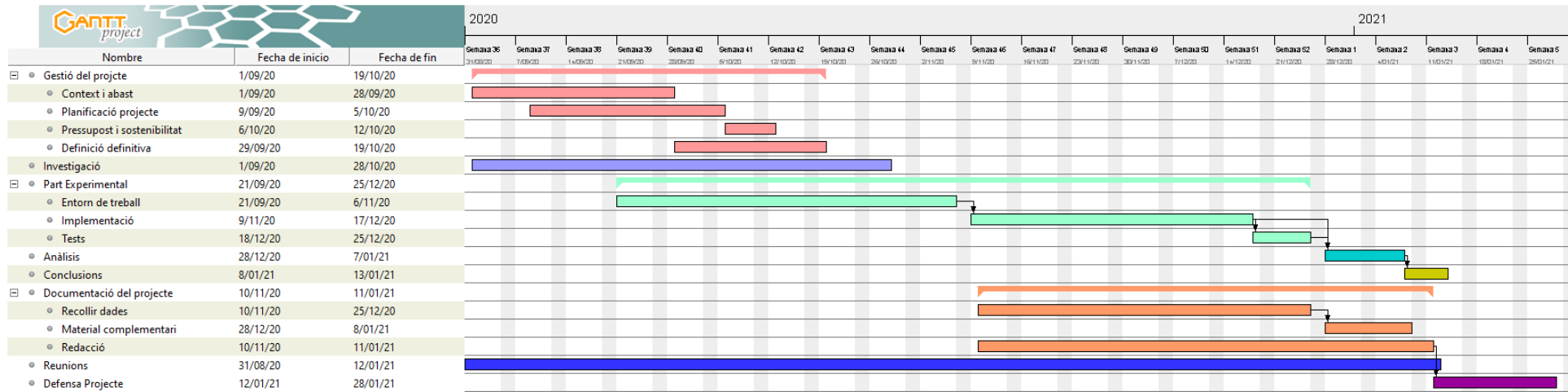


Figura 2.2: Diagrama de Gantt amb desviacions.
 Diagrama de Gantt de la realització del projecte després de les desviacions. Elaboració pròpia.

Capítol 3

Pressupost i sostenibilitat

En aquesta secció és descriure el pressupost de la realització del treball. S'ha desglossat les despeses segons la seva naturalesa. També es realitza un estudi de la sostenibilitat del projecte. S'estudia si el treball és viable en l'entorn real on treballem amb un context econòmic i s'ha de considerar l'impacte ambiental.

3.1 Pressupost

La viabilitat d'un projecte es veu fortament influenciat pel pressupost d'aquest. Segons els diners que hi hagin necessitat per realitzar el mateix, el resultat és més atractiu o menys. L'abast del projecte també es veu limitat molts cops per aquest factor. Es consideren les despeses totals del projecte sigui directament o indirectament. Primerament cal mencionar que el treball s'ha realitzat en un entorn acadèmic, per tant, el pressupost dista del que es realitzaria en un entorn empresarial pels actors que es veuen involucrats.

3.1.1 Agent

Els actors que es veuen involucrats en el treball són mencionats anteriorment. El temps de l'autor es comptarà com la compensació que marca el mercat laboral ajustant lo a les hores treballades al dia i l'experiència. Un professional com aquest pot cobrar entre 36.000€ i 39.000€[1]. Tenint en compte que es treballen 4 hores al dia durant gairebé 5 mesos, el total surt una quantitat aproximada a 7.500€.

3.1.2 Material

En els capítols anteriors s'ha mencionat en detall les eines que s'han fet servir per realitzar aquest treball. En la llista tenim diversos equips informàtics: un equip personal i un equip d'oficina. En la següent taula tenim una recopilació de característiques que ens ajudaran a calcular el cost d'aquest.

Equip	Preu	Preu residual	Amortització	Cost (5 mesos)
Personal	950€	300€	8 anys	34€
Oficina	1300€	400€	8 anys	47€
Total	2250€	700€		81€

Taula 3.1: Cost equips informàtics.

Electricitat

El cost mitjà d'electricitat per llar és de 56,3€ per mes [9] . Durant els pròxims cinc mesos la quantitat augmentarà a 251.5€.

Aigua

El preu mitjà de la factura de l'aigua a Barcelona és de 23.16€ per més[2]. Durant els pròxims 5 mesos la quantitat augmentarà a 115.8€.

Lloger

Disposar d'una habitació a Barcelona és d'aproximadament 400€ per mes. Durant els pròxims 5 mesos la suma augmentarà a 2.000€.

Internet

Disposar de connexió a internet surt aproximadament a 29.99€ al mes . En cinc mesos la quantitat augmenta a 149.95 €.

3.1.3 Cost total

En aquest apartat es fa una recopilació de totes les despeses del projecte per donar una visió més global de la seva viabilitat. En la següent taula es recopilen les dades mencionades anteriorment.

Concepte	Cost
Personal	7.500€
Informàtica	81€
Electricitat	251,5€
Aigua	115,8€
Internet	149,95€
Lloguer	2.000€
Total	10.098,25€

Taula 3.2: Recopilació de despeses.

3.1.4 Control de la gestió

És molt probable que el pressupost anterior es vegi modificat per diverses raons com les mencionades en la secció de riscos. En cas de necessitar fer modificacions, s'ha de tornar a calcular les quanties necessàries per acabar el projecte. Aquestes modificacions són igual d'importants que el pressupost inicial donat que s'ha de tornar a considerar la viabilitat d'aquest. Per sort, en aquest

treball els costos són orientatius i no presenten un risc real pel projecte.

A continuació hi ha un llistat de mètriques a tindre en compte per calcular la desviació de cada tasca un cop ha finalitzat.

- **Desviació del cost de recursos humans:** és el cost produït per les hores extres que realitzen els actors del projecte. Es calcula amb la següent fórmula:

$$D.C.R.H = \sum (Hores_previstes - Hores_extres) * Remuneració_per_hora \quad (3.1)$$

- **Desviació cost Hardware:** en el pressupost tenim en compte l'amortització dels equips informàtics que s'utilitzen. En de la incorporació de nous equips a causa de fallades també s'ha de tindre en compte l'amortització d'aquests. En cas de modificar un dels equips mencionats en el pressupost s'ha de reajustar les amortitzacions.

$$D.C.H = \sum (Duració_projecte - Duració_extra) * Amortització \quad (3.2)$$

- **Desviació despeses mensuals:** Llum, aigua, Internet i lloguer són uns despeses mensuals que no varien segons les hores de treball diàries. Per tant, només es considera una extensió de la data límit com a cost extra a considerar.

$$D.D.M = (Electricitat + Aigua + Internet + Lloguer) * Extensió_Deadline \quad (3.3)$$

3.1.5 Desviació del pressupost

En la secció 2.4 s'han descrit les desviacions en el pla temporal del projecte. Això és fet usual durant la realització de projectes. En tot cas, s'ha replanificat la distribució del treball per complir les dates d'entrega. 104 hores extres s'han necessitat per complir els objectius establerts. En aquesta secció s'estudia l'impacte que aquest fet ha tingut en el pressupost fet anteriorment. Es considera que aquestes hores equivalen a 1,5 mesos extres.

- **Desviació del cost de recursos humans:** el salari que rebria un professional del sector pel treball seria de 2.250€ segons l'equació 3.1.
- **Desviació cost Hardware:** aquesta desviació introdueix un cost addicional de 24,3€ calculat amb l'equació 3.2.
- **Desviació despeses mensuals:** aquesta desviació introdueix un cost addicional de 84,45€ calculat amb l'equació 3.3. En aquest cas, només s'ha considerat el consum d'electricitat requerit per fer funcionar els equips informàtics durant les hores extres.

En la següent taula es recopilen la informació de les desviacions.

Concepte	Cost
Pressupost	10.098,25€
Desviacions	2.358,75€
Total	12.457€

Taula 3.3: Desviacions.

3.2 Sostenibilitat

La sostenibilitat d'un projecte està indicat per diversos factors. És un concepte molt important a tenir en compte en durant la planificació. Un projecte sostenible té en compte la integració del mateix en el context especificat sense produir danys o inconvenients majors. Un projecte sostenible és aquell que és capaç d'integrarse en fase de producció de forma efectiva. S'ha de considerar l'impacte econòmic, social i ambiental de les conseqüències del treball. D'aquesta manera, hem de pensar en el cas que escalés de forma prominent o altres actors facin projectes amb impactes similars. Doncs, l'objectiu és minimitzar els recursos utilitzats perquè es puguin fer servir en futurs treballs. Durant aquesta secció, es respondran diverses preguntes que reflectiran l'impacte de les diferents dimensions del projecte.

3.2.1 Dimensió econòmica

Es reflecteix els costos necessaris per completar el projecte?

En la secció 3.1 es presenta el pressupost fet servir per realitzar el projecte. Es tenen en compte els costos derivats d'aquest. Des dels equips informàtics, software, personal, etc. S'ha volgut reflectir al detall i de forma realista les despeses realitzades. El cost del personal estimat dista del que es faria servir en una empresa deguda que en un entorn acadèmic els directors tenen un cost equivalent a la matrícula del projecte. L'autor tampoc té un cost pel projecte, en canvi, un dissenyador hardware suposaria un gran cost a causa de les bones valoracions que tenen en el mercat laboral. Tenint en compte tot això, el pressupost té en compte aquest factor i s'ajusta a la realitat.

En cas de voler traslladar el projecte a un entorn empresarial s'hauria d'ajustar el cost del personal al qual es veurà incrementat de forma important. Normalment, es contracta a un equip de desenvolupadors els quals es divideixen en diversos rols: tester, dissenyador hardware, coordinador... En aquest cas, tots aquest rol són realitzats per l'estudiant.

Com se solucionen actualment el problema econòmics respecte al problema que s'adreça?

Actualment el problema és adreçat pel projecte DRAC, el qual disposa de grans recursos. Equips d'enginyers que solucionen el problema amb diverses aproximacions de la solució. És un projecte amb consorci de la Universitat Politècnica de Catalunya (UPC), Universitat Autònoma de Barcelona (UAB), Universitat de Barcelona (UB), Universitat Rovira i Virgili (URV) i el Barcelona Supercomputing Center (BSC). Compte amb el cofinançament de 2.000.000€ de la European Union Regional Development Fund i amb el suport de la Generalitat de Catalunya[7].

Com millorarà la teva solució els problemes econòmics respecte a les altres solucions existents?

Les aproximacions proposades en el treball ajudaran al fet que el processador resultant sigui més eficient. Tant estalviant en temps de càlcul o en energia consumida. Per aquesta raó, la fabricació del xip a gran escala seria menys costós.

3.3 Dimensió ambiental

S'ha estimat l'impacte ambiental del projecte?

Referent a l'impacte ambiental del projecte, no es consumeixen materials per la realització d'aquest. Els experiments són simulacions de hardware igual que els mateixos dissenys dels circuits. El consum elèctric tampoc és el factor més contaminant del treball. Així i tot, els equips no consumeixen una quantitat massa elevada com que són ordinadors personals i a més no aprofiten el 100% dels recursos pel que el consum també es redueix considerablement.

S'ha planificat el projecte per minimitzar el seu impacte, per exemple, minimitzant els recursos?

Donat que el projecte consumeix recursos molt limitats, no s'ha considerat minimitzar els recursos utilitzats. No ha estat necessari comprar material nou o consumir els. Per tant, s'ha prioritzat la comoditat i l'eficiència. Com a alternativa, es podria haver utilitzat els equips informàtics de la UPC que ja disposen de tot el necessari per realitzar la part tècnica.

Referent a la vida útil: Com se soluciona actualment el problema que vols resoldre (estat de l'art)?

Actualment el paradigma de disseny és utilitzar processadors que utilitzen operacions exactes per calcular resultats que potser accepten marge d'error. Aquests processadors intenten ser més eficients mitjançant tècniques com la segmentació o replicació.

Com la solució millorarà el medi ambient respecte a les solucions ja existents?

Les propostes de processador o accelerador permet fabricar equips més ràpids i eficients en energia. Sacrificant precisió en el tractament de dades que no afecten el resultat final de les operacions per l'usuari. En aplicacions on hi intervenen sensors o usuaris humans, es pot assumir cert grau d'error.

3.4 Dimensió social

Quins objectius es creu que es compliran com a realització personal?

Durant la realització del treball, l'autor dissenyarà i realitzarà un experiment sobre dissenys hardware. Aquesta és la principal motivació. Establir un criteri d'anàlisi que es pugui assimilar als comparar a les tècniques que s'utilitzen avui en dia en la investigació d'aquest àmbit.

Com es soluciona actualment el problema que es vol adreçar?

Actualment, la comunitat científica investiga i comparteix els resultats dels seus experiments. Els articles són una font fiable i accessible d'informació en aquest àmbit. Però, les tecnologies més punteres són desenvolupades per companyies privades i no comparteixen els seus resultats.

En quins àmbits la teva solució millorarà la qualitat de vida respecte a les altres solucions?

Un dispositiu inspirat en les aproximacions que es realitzen, ofereix al consumidor final un funcionament similar consumint menys energia i ocupar menys espai. És el camí que s'ha de seguir per un consumisme responsable i sostenible.

Una altra part de la societat que es veurà beneficiada són els investigadors que llegeixin l'estudi i utilitzin les conclusions per formar-se i utilitzar el en el seu treball.

És una necessitat real realitzar el projecte?

La tendència de la societat actual és d'utilitzar cada dia més tipus de dispositius electrònics. El mercat augmenta la seva oferta alhora que els seus consumidors augmenten la demanda. Es pretén solucionar problemes que no tenen prioritat per la societat però que sí que són greus i urgents.

Capítol 4

Verificació i tests

Aquesta secció explica els mètodes de verificació i testatge que s'han utilitzat durant el projecte. En la secció 1.5 es menciona la necessitat de comprovar el correcte funcionament dels dissenys. Donat que els circuits són creació específica per les necessitats del treball, han de recolzar-se en un sistema sòlid de verificació. Seguidament s'explica: el sistema de verificació de components i model base, el sistema de creació de tests i l'automatització de l'anàlisi de mètriques.

Mencionar la importància del treball realitzat en aquesta etapa. S'han invertit una gran quantitat d'hores en dissenyar sistemes que siguin òptims en temps d'ús i que produeixin resultats fiables. S'ha automatitzat la comparació de models mitjançant dos scripts que s'explicaran més endavant i s'ha dissenyat un procediment de verificació de circuits. Per últim es veu com s'han obtingut les mètriques que s'utilitzen en l'anàlisi i comparatives.

4.1 Verificació de components

El model base és fragmentat en el disseny de diversos components que tenen tasques específiques. Aquesta forma de descriure el circuit es considera una bona pràctica de programació. Cada component pot ser descrit en un fitxer el qual no ha d'ocupar més espai del que es pot visualitzar amb l'eina d'edició que utilitzem. Si això no és possible, s'ha de considerar encapsular circuit en diferents mòduls. Un disseny modular és més efectiu en espai de codi, claredat de lectura i verificació d'aquest.

Aquesta ha estat la filosofia de verificació seguida en aquest treball. S'ha començat per dissenyar les instàncies més bàsiques, tal com Full Adders o sumadors de diversos bits. Són components senzills que es programen ràpidament. També va servir com a presa de contacte amb l'entorn de treball. Un cop s'escriu el codi complet, s'ha de compilar amb Quartus II.

Per això es crea un projecte en Quartus II per cada component. Gràcies al "Project Wizard" és un procés senzill. El nom del projecte ha de coincidir amb el nom que li hem donat al "Top Entity", que és el component en qüestió. Per poder utilitzar diferents mòduls en el "Top Entity", hem d'incloure els fitxers dels mateixos en el projecte. També hem d'incloure components dels mateixos components.

El següent pas del "Project Wizard" ens demanarà un el model d'una FPGA per utilitzar en el projecte. No és necessari escollir un model si la intenció és simplement comprovar la sintaxi del

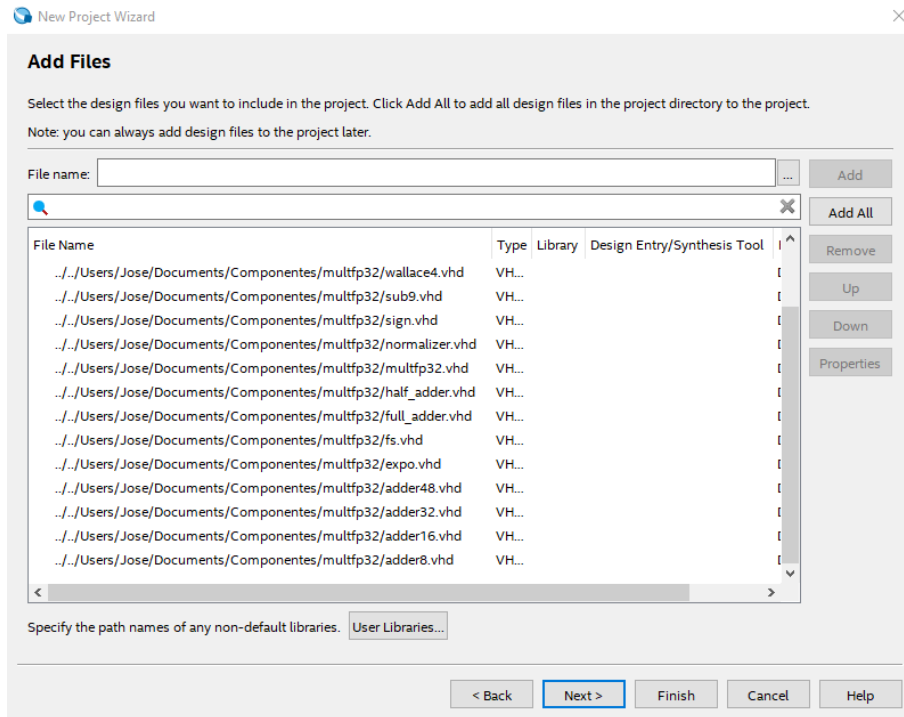


Figura 4.1: Add Files.

Interfície de Quartus II per afegir fitxers fonts de codi. Elaboració pròpia.

model o inclús realitzar proves amb Model Sim. Per l'altra banda, sí que és necessari per simular la síntesi del circuit de forma idònia. Doncs Quartus II necessita les característiques del xip destí. El model escollit és el Cyclone IV E: EP4CE115F29C7 a voltatge d'1,2 V. Aquest és el mateix model que utilitzen les Development Boards Altera DE2-115 que s'utilitzen en la FIB en assignatures com Projecte d'Enginyeria de Computadors (PEC) on es dissenya una CPU. Per tant és una elecció adient donat la familiaritat que té l'autor amb el xip.

Mencionar que per utilitzar la configuració de qualsevol xip FPGA, és necessari descarregar i instal·lar "packages" corresponent a la família del xip. Aquests "packages" estan disponible gratuïtament a la pàgina web d'Intel.

Un cop finalitzats amb el "Project Wizard", podem procedir a realitzar un "Analysis Elaboration", el seu output indica si hi ha alguna errada de sintaxi. Una altra eina que ens ofereix Quartus II és el "Simulation WaveFront Editor". Ens servirà per editar el fitxer amb extensió "University Program VWF" associat al projecte. D'aquesta manera podem configurar senyals d'entrada per alimentar el circuit en les simulacions. És una eina adient donat que podem generar de forma aleatòria inputs aleatoris, senyals de rellotge, etc.

Seguit aquests passos, es poden començar les simulacions del component a testejar. En la figura 4.3. es pot observar l'output d'una simulació amb l'eina Model Sim pel component Exponent de la secció 5.3. En la "Wavefront" veiem una representació temporal de la simulació i els valors del senyal que es consideren interessants, en aquest cas Inputs i Outputs. Si és necessari, es poden afegir o retirar senyals d'aquest. Podem verificar el seu correcte funcionament comprovant que els outputs són els esperats.

Aquest és el procés utilitzat per verificar els components tant dels dissenys del model base com dels models aproximats.

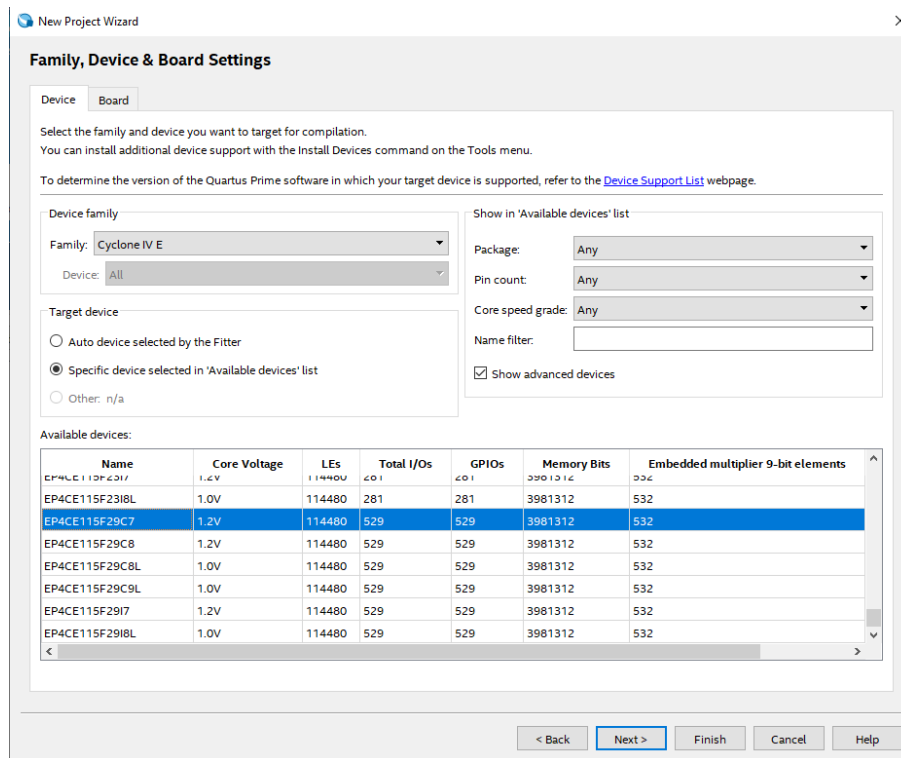


Figura 4.2: Board Settings.

Interfície de Quartus II per escollir el xip FPGA assignat al projecte. Elaboració pròpia.

4.2 Eines de Quartus

Els dissenys que s'han compilat i verificat adientment es poden sotmetre a les eines de Quartus II per calcular temps crític i consum energètic. Aquestes eines són Timing Analyzer i Power Analyzer. En aquesta secció s'explica el procés que s'ha utilitzat per analitzar components i models.

4.2.1 Timing analyzer

Timing analyzer requereix la compilació de Quartus II i del fitxer SDC del model. Un fitxer SDC conté limitacions de temps i àrea del disseny. És una eina amb gran quantitat d'opcions, en aquest cas no es necessitaran la majoria.

La comanda "Report Path" ens permet ajustar un seguit de paràmetres. Es pot especificar quins nodes d'inici i de final es vol estudiar el temps del camí, així com els nodes intermedis. Per defecte, només reportarà el camí amb més retard, però també es pot indicar la quantitat de camins a reportar. Un cop s'han escollit els paràmetres adients, Timing analyzer ens indicarà els camins de dades que ha analitzat i els seus corresponents retards. De tots aquests, l'eina destaca el que ha tingut un retard més gran.

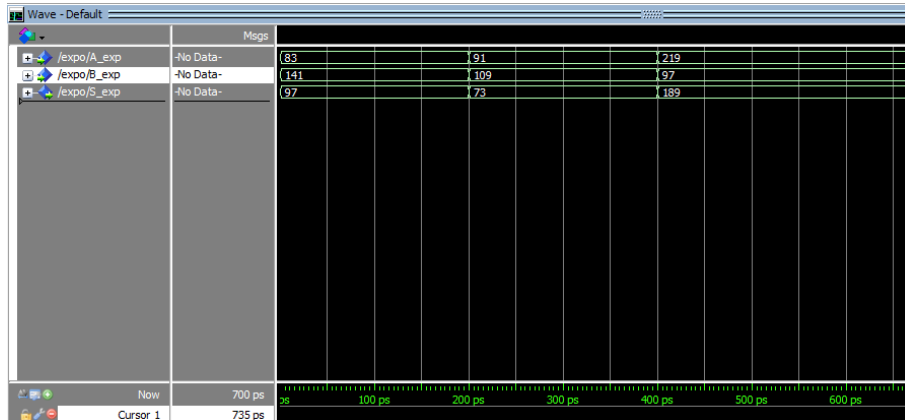


Figura 4.3: Simulació ModelSim.

Captura de ModelSim durant el test del mòdul Exponent. Elaboració pròpia.

4.2.2 PowerPlay Power Analyzer

PowerPlay Power Analyzer és una eina que et permet fer una estimació precisa del consum d'un dispositiu. Per adquirir el màxim nivell de fiabilitat en els resultats s'han de proporcionar certa informació. Segons l'etapa del disseny, es poden fer diferents estimacions. PowerPlay Early Power Estimator (PowerPlay EPE) és una eina complementària a PowerPlay Power Analyzer, que ens permet calcular el consum del circuit quan aquest no està finalitzat i de forma menys precisa. En aquest projecte, s'utilitza exclusivament el Power Analyzer pel fet que els dissenys es finalitzen amb rapidesa i una primera estimació del consum no seria d'utilitat.

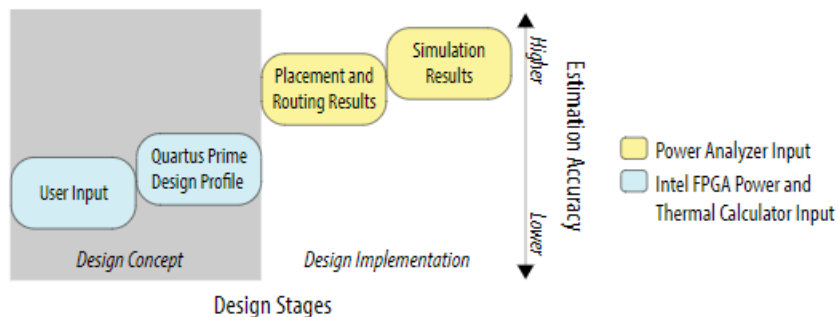


Figura 4.4: Etapes Power Analyzer.

Representació de les etapes d'un projecte i l'estimació del consum. Reproduced from [18].

Un cop s'ha compilat el projecte de Quartus II, Power Analyzer pot calcular el seu consum. La quantitat de paràmetres que es poden configurar és molt alta. D'aquesta manera es pot indicar a l'eina les condicions i el comportament que els dissenys tindran. Aquesta és informació necessària per fer un càlcul acurat. Un paràmetre important és el "Toggle Rate" que indica el percentatge en la que un senyal canvia durant el temps. Per defecte té un valor de 12,5%. Una altra opció per indicar com els senyals fluctuen de valor és generar un fitxer VCD amb els resultats de simulacions. El fitxer VCD registrarà tots els canvis en els senyals dels dissenys durant una simulació. En aquest projecte s'ha realitzat una simulació especial per cada model per obtenir aquest valor. Cal mencionar que registrar aquesta activitat és un procés costós i que genera fitxers de molt pes. Tenint en compte això, si s'afegeix aquest registre a la simulació de 150.000 operacions, la simulació s'allarga dels 30 minuts a les 2 hores aproximadament i generant un fitxer VCD d'aproximadament

2 GB. Per evitar això, s'ha decidit realitzar una segona simulació amb només 1.000 operacions. Aquestes operacions corresponen a les 1000 primeres operacions generades pel script de la secció 4.3.1 i s'emmagatzema en el fitxer Random1000.txt.

Per últim, Power Analyzer té una secció de configuració on s'indica quines són les solucions de refrigeració i la temperatura en la qual opera el dispositiu. Per aquest projecte s'ha escollit una temperatura ambient de 25 graus. El sistema de refrigeració escollit ha estat un corrent d'aire de 200 LfPM (Linear Feet per Minute) que equival a aproximadament 1,016 m/s i sense dissipador. Amb aquests paràmetres, Power Analyzer calcula el consum total mitjançant la suma dels següents consums:

- **Core Dynamic Thermal Power Dissipation:** potencia consumida en els canvis d'estat del model.
- **Core Static Thermal Power Dissipation:** potencia consumida per mantenir l'estat del model.
- **I/O Thermal Power Dissipation:** potencia consumida pels capacitadors per alimentar les entrades del model.

Un cop tots els paràmetres s'ajusten a les condicions del treball, es pot processar el càlcul del consum.

4.3 Sistema d'automatització de tasques

Durant el procés de verificació i tests, hi ha diverses tasques monòtones. Uns exemples són triar operands per fer les operacions, comparar resultats entre diferents models, fer percentatges, etc. Per això s'ha dissenyat diverses eines per facilitar aquestes tasques. En aquest capítol s'expliquen les característiques de les mateixes i quins són els requisits que hi han de complir.

4.3.1 Generació d'operands

Els models aproximats necessiten una sèrie de números amb els quals fer operacions per a posteriori estudiar quin ha estat el seu comportament. Per això, s'ha generat 150000 parelles de nombres aleatoris. Cada parella representa una operació. Tots els models utilitzaran la mateixa sèrie per assegurar la coherència dels resultats. S'ha triat el llenguatge Python per programar el script que farà aquest treball. La facilitat d'ús i la seva clara sintaxi han estat les raons per utilitzar-lo.

El script no només ha de generar les parelles d'operands, sinó que també ha de complir un certs requeriments pel correcte funcionament amb la resta de mòduls com el Testbench escrit en VHDL. Aquests requeriments són:

- **Persistència:** tots els models han de ser testejats amb els mateixos operands. Per aquesta raó, és necessari un mètode de persistència per l'output. S'ha escollit un fitxer pla de text.
- **Format IEEE-754:** els models han estat dissenyats seguint una arquitectura que segueix els estàndards del IEEE-754. Representació binària.
- **Números normals:** els operands han de ser números normals segons l'estàndard.

- **Resultat normal:** s'ha de descartar les parelles d'operands que després de l'operació no produeixin un resultat normal. També s'han de vigilar Overflows. Una funció comprova els exponents generats i que el seu resultat estigui en el rang esperat.

Un cop s'han generat els operands correctament, el fitxer es pot fer servir com a entrada del Testbench dels models.

4.3.2 Testbench

Un Testbench es tracta d'una arquitectura que és especialment dissenyada per testejar un component en concret. El Testbench no té entrades ni sortides. Al seu interior només es tracta d'una instància del component a examinar. A cada input i output del mateix se li assigna un Signal VHDL. Mitjançant aquests Signals, es pot alimentar de forma simultània totes les entrades del circuit i registrar en les sortides que genera. Gràcies a simulacions amb l'eina Model Sim, s'estudia el comportament dels estímuls en una línia temporal.

En aquest cas, també s'ha implementat un sistema de persistència pels outputs dels estímuls. Això no ha fet que l'arquitectura del banc de proves sigui complex, perquè es pot dividir en dues parts:

- **Part concurrent:** aquí es troba la instància del component a analitzar. Tots els seus ports són connectats directament a un Signal.
- **Procés:** mitjançant un procés de VHDL, es pot llegir dades d'un fitxer (generat pel script de la secció 4.3.1) i assignar aquests valors a senyals. Aquesta és la forma amb la qual els operands binaris generats amb el script anterior alimenten els nostres models. Seguint el mateix mètode també es poden escriure Signals en fitxers i així registrar els outputs del model. Tot això ha d'anar sincronitzat per un senyal de rellotge que asseguri que hi hagi interferències entre operacions consecutives.

Un cop es fa una execució d'aquest banc de proves, el resultat és emmagatzemat en un fitxer pla de text, on cada línia és un número binari de 32 bits en format IEEE-754. La línia "n" correspon als productes del número de la línia "n" del fitxer d'entrada.

4.4 Comparació de resultats

Després de tot el procés descrit anteriorment, s'ha generat un fitxer per a cada model amb els resultats de les 150.000 operacions utilitzada com entrada. Lògicament, cada model de circuit aproximat ha calculat amb les aproximacions aplicades. El següent pas és calcular l'error introduït pels models aproximats.

Un altre script en Python ha estat dissenyat per acomplir aquesta tasca. S'han afegit noves funcionalitats com convertir les cadena de caràcters binaris en números en coma flotant. Necessari per comparar resultats de models diferents, però també és útil per transformar operacions que s'han utilitzat. D'aquesta forma es pot utilitzar una altra referència: el producte calculat per Python.

El model base ha estat dissenyat especialment per tindre un punt de partida per les aproximacions, però també s'ha considerat adient testejar aquest component. Per això s'ha utilitzat els càlculs dels productes fets per Python.

Durant el càlcul dels errors es té en compte els diversos casos a vigilar:

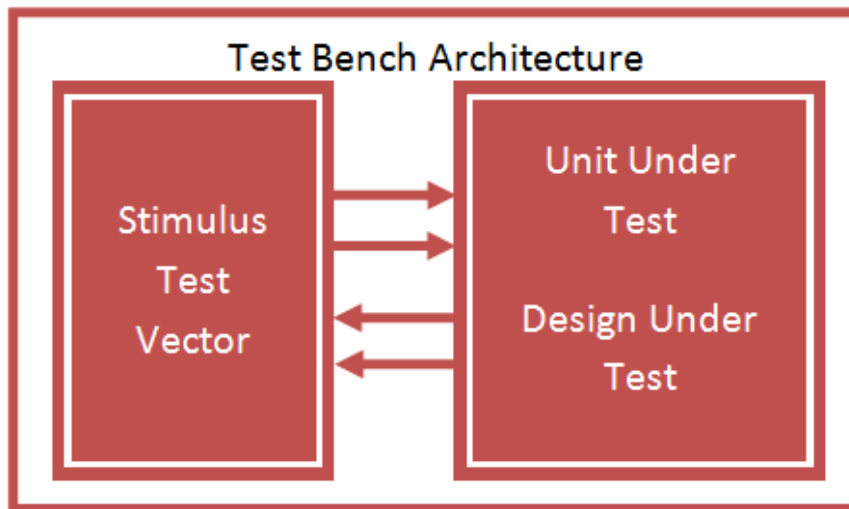


Figura 4.5: Test Bench.

Diagrama representatiu de l'arquitectura d'una unitat Test Bench. Reproduced from [20].

- **Resultat referència no sigui "0"**: com s'explica en la secció 1.1.3, en el moment de calcular la proporció d'error, s'ha de vigilar les divisions entre 0.
- **Error en el format**: el producte de dos números normals és un altre número normal i això és el que espera el script. Encara que en la generació dels operands s'han descartat molts casos d'Overflow o Underflow, s'ha de descartar resultats que no siguin números normals. Especialment en els models aproximats.

El programa calcula per cada operació: l'error absolut i el percentatge d'error. Per últim, de tots els resultats calcula: Percentatge d'error mitjà, percentatge d'error màxim i percentatge d'error mínim i els quantils: 0.25, 0.50 i 0.75. Tots aquests valors ens ajudaran a saber la magnitud de l'error introduït durant les aproximacions i la seva distribució.

En el següent diagrama podem veure una representació gràfica d'aquest procés.

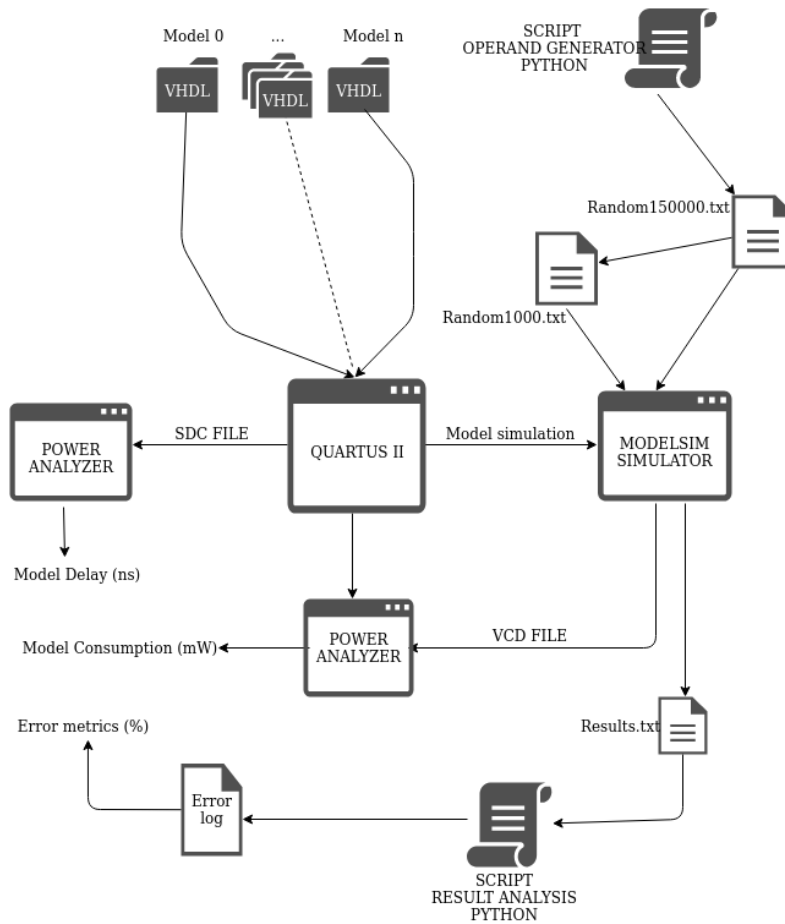


Figura 4.6: Procés de Verificació i Tests.

Diagrama del flux de dades que s'ha utilitzat en l'anàlisi dels circuits durant el projecte.
Elaboració pròpia.

Capítol 5

Model Base

En aquesta secció es descriu el disseny del circuit multiplicador que servirà de referència per les aproximacions que s'aplicaran després. Tots els mòduls són descrits amb VHDL (VHSIC HDL o Very High Speed Integrated Circuits Hardware Description Language) mitjançant l'eina Quartus II. Aquest circuit es pot implementar de diverses formes. Tot i això, l'arquitectura s'ha escollit seguint uns criteris de simplicitat i tenint en compte les modificacions que s'han de realitzar. De les mètriques obtingudes d'aquest primer disseny, mesurarem els efectes de les tècniques d'aproximació futures. També es descriu requisits que ha de complir el mateix com els seus derivats.

5.1 Característiques del circuit

El circuit implementa un multiplicador de coma flotant que segueix l'estàndar IEEE-754. Com hem mencionat abans, aquest document defineix diverses regles per a diferents arquitectures. Aquest model és únicament de 32 bits, per tant, l'estudi es farà en arquitectures de 32 bits. Un altre detall característic és que es tracta d'un circuit combinacional. A diferència del seu antagonista, un circuit seqüencial, aquest tipus no disposa de registres. Per això no necessita d'un senyal de rellotge per sincronitzar els possibles estats del circuit. El seu estat només depèn de les entrades que alimenten el circuit.

Un altre avantatge que presenten els circuits asíncrons per aquest estudi, és el fet que el retard coincideix amb el camí crític del circuit. Per tant, comparar els retards dels diferents models i realitzar les simulacions és més senzill. Per aquestes mateixes raons, també s'ha decidit no incloure excepcions causades per l'aritmètica de coma flotant ni tampoc un sistema d'arrodoniment com s'ha mencionat en la secció 1.1.3. Encara de no disposar d'aquest mòdul, el model base ha de disposar de la màxima precisió perquè l'anàlisi sigui pròxim a la realitat.

Les entrades consisteixen en vectors de 32 bits que segueixen els valors de números normals de l'estàndar utilitzant el mètode de la secció 4.3.1. Aquest mateix mètode també a segura que el resultat sigui un número normalitzat. El producte també ha de complir el format IEEE-754. Aquest model és una implementació hardware de la fórmula que algebraicament es descriu de la següent forma:

Amb els operands:

- $X = (-1)^{s1} * 1.mant1 * 2^{e1}$
- $Y = (-1)^{s2} * 1.mant2 * 2^{e2}$

Es calcula el producte:

$$Z = (-1)^{(s1 \oplus s2)} * (1.mant1 * 1.mant2) * 2^{(e1+e2)-127} \quad (5.1)$$

5.2 Mòdul Signe

El mòdul signe és l'encarregat de calcular el signe del resultat, positiu o negatiu. La seva estructura és senzilla.

- **Input i preconditions:**

- **A_sign:** bit que representa el signe del factor anomenat A. Posició 31 en el primer operand d'entrada.
- **B_sign:** bit que representa el signe del factor anomenat B. Posició 31 en el segon operand d'entrada.

- **Output i postconditions:**

- **S_sign:** bit que representa el signe del producte dels factors A i B. Valor de la posició 31 de l'output del model.

La implementació consisteix en una simple porta lògica XOR que rep directament els inputs i genera l'output. El retard del circuit és de 7.805 ns.

5.3 Mòdul exponent

Aquest mòdul calcula mitjançant dues operacions aritmètiques l'exponent corresponent al resultat. Els exponents dels operands A i B se sumen segons les propietats de l'aritmètica del producte de potències de la mateixa base [11]. A continuació, es resta el valor numèric 127 atès que en l'operació anterior els dos operands tenien un format amb excés de 127.

- **Input i preconditions:**

- **A_exp:** 8 bits que representen en un nombre natural l'exponent de l'operand A en excés de 127. Correspon al rang de bits amb posicions des de 30 a 23 del primer operand.
- **B_exp:** 8 bits que representen en un nombre natural l'exponent de l'operand B en excés de 127. Correspon al rang de bits amb posicions des de 30 a 23 del segon operand.

- **Output i postconditions:**

- **S_exp:** 8 bits que representen en un nombre natural l'exponent del producte dels factors A i B. Correspon al rang de bits amb posicions 30 a 23 del resultat sense normalitzar.

La implementació consisteix en un sumador de 8 bits amb propagació del carry. El resultat de 8 bits i el carry es concatenen i alimenten l'entrada d'un restador de 9 bits juntament amb la codificació binària del número 127. L'estructura es mostra en la figura 5.1. El camí crític del circuit és des de el bit 0 de qualsevol dels dos operands fins en el bit de més pes de l'output. El retard del circuit és de 11.551 ns.

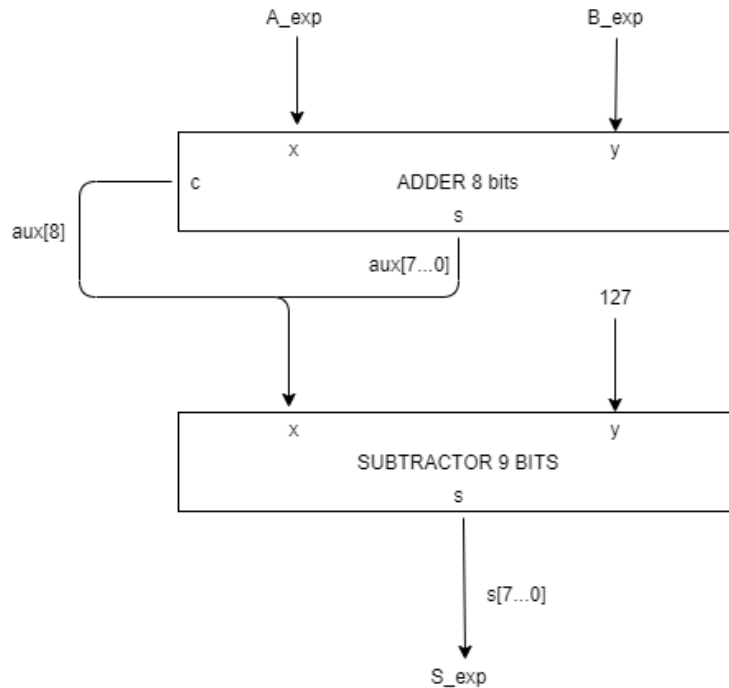


Figura 5.1: Mòdul Exponent.
Diagrama del mòdul Exponent. Elaboració pròpia.

5.4 Wallace Tree de 24 bits

Aquest mòdul és el cor de la unitat funcional. La seva tasca és calcular el producte de les dues mantisses. Es tracta de l'etapa que més energia es consumeix i més retard genera. L'estructura general es basa a calcular diversos productes parcials, en concret 4, i fer les sumes d'aquests.

- **Input i preconditions:**

- **A:** 24 bits que representen la mantissa i el bit explícit. Els bits de mantissa són les posicions de l'operand A de les posicions 22 a 0. El bit de més pes es tracta d'un 1 implícit corresponent a la representació normalitzada de nombres en coma flotant.
- **B:** 24 bits que representen la mantissa i el bit explícit. Els bits de mantissa són les posicions de l'operand B de les posicions 22 a 0. El bit de més pes es tracta d'un 1 implícit corresponent a la representació normalitzada de nombres en coma flotant.

- **Output i postconditions:**

- **prod:** 48 bits que representen el producte dels factors A i B.

S'utilitzen diversos mòduls Wallace Tree més petits tal com es mostra a la figura 5.2 i després s'agreguen els resultats. Els dos operands es divideixen per la meitat i es calcula totes les combinacions de parts baixes i altes. El producte de les 4 combinacions és suma en tres sumadors parcials. El seu retard és de 54,209 ns.

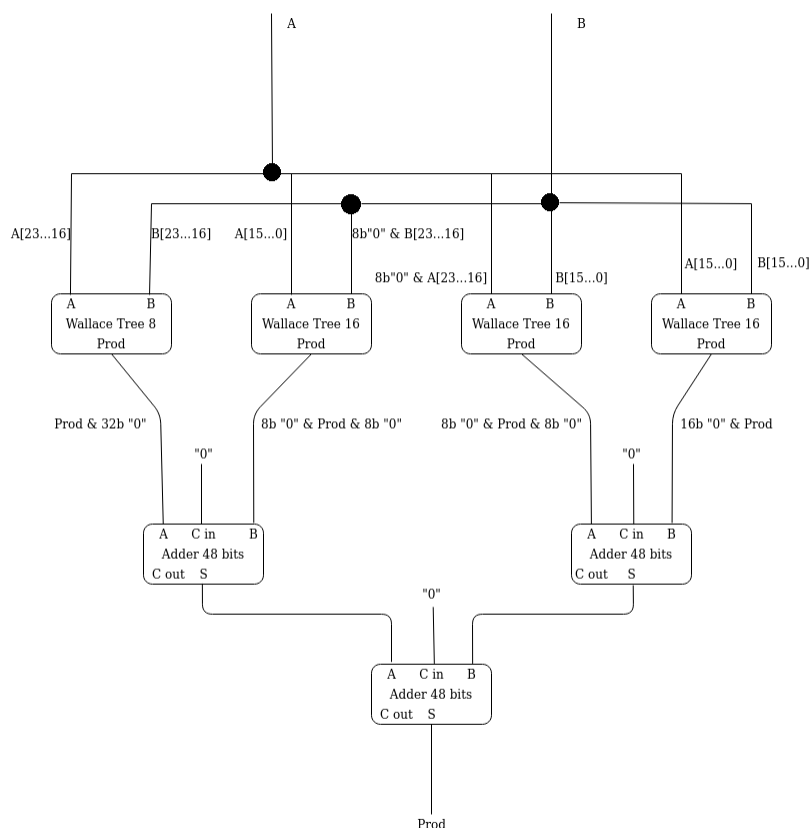


Figura 5.2: Mòdul Wallace Tree.
Diagrama del mòdul Wallace Tree de 24 bits. Elaboració pròpia.

5.5 Mòdul normalitzador

El mòdul normalitzador és l'encarregat de donar format al resultat del producte de mantisses. Durant aquest procés, l'exponent també es veu modificat perquè així l'output total segueixi la notació científica. Aquest mòdul ha estat dissenyat seguint les directives de la font [8]. En aquest treball[8], els autors dissenyen un multiplicador de coma flotant que compleix els requisits descrits en la secció 5.1. Tot i això, el mòdul normalitzador ha estat l'únic que s'ha inspirat en aquest document pel mateix benefici del projecte. Doncs els dissenys que allà es presenten són optimitzats. En el cas del normalitzador, descrit en la figura 5.3, es volia un disseny senzill d'implementar i que segueixi l'estàndar IEEE -754.

- **Input i preconditions:**

- **Res:** 48 bits que representen el producte de mantisses de dos números normalitzats. Equival a la sortida del mòdul wallace 24 descrit anteriorment.
- **EXP_in:** 8 bits que representen l'exponent del resultat en excés de 127. Equival a la sortida del mòdul exponent.

- **Output i postconditions:**

- **EXP_n:** 8 bits que representen l'exponent del resultat en excés de 127.
- **RES_n:** 24 bits que representen la mantissa normalitzada amb el bit ímplicit.

Partint d'una de les premisses de [8], el resultat de multiplicar dues mantisses ja normalitzades assegura que almenys una de les dues posicions de més pes tingui un 1. Fet que simplifica la normalització, perquè reduïm la casuística a 2 escenaris segons el valor del bit de més pes de Res:

- **Cas valor 1:** el valor de RES_n és igual als bits de Res entre les posicions 47 a 24. És necessari sumar 1 a l'exponent EXP_{in} per normalitzar-lo.
- **Cas valor 0:** el valor de RES_n és igual als bits de Res entre les posicions 46 a 23. L'exponent normalitzat correspon directament al valor de EXP_{in}.

Destacar que en qualsevol dels dos casos, RES_n sempre té un 1 en la posició de més pes. Conceptualment correspon al bit explícit dels números en IEEE-754. El seu camí crític comença a l'input Res en el seu bit de més pes i acaba en el senyal de sortida EXP_n en el bit amb posició 4. El retard és de 9.650 ns.

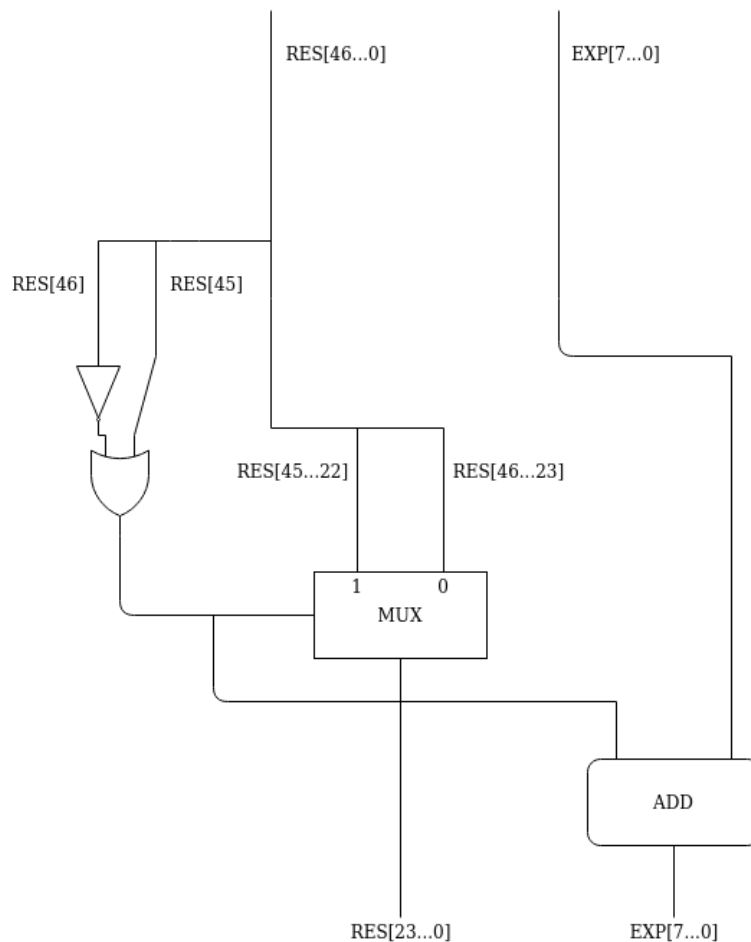


Figura 5.3: Mòdul Normalitzador.
Diagrama del mòdul Normalitzador. Elaboració pròpia.

5.6 Multiplicador Coma Flotant 32 bits

El mòdul multiplicador coma flotant 32 bits o també referenciat durant el document com a model base, és l'encarregat de la interconnexió dels mòduls descrits anteriorment. La seva estructura és senzilla, descrita en la figura 5.4, i la seva principal tasca és gestionar el correcte ús dels mòduls.

- **Input i preconditions:**

- **A:** operand de 32 bits que segueixen l'estàndard IEEE-754 per a números normalitzats.
- **B:** operand de 32 bits que segueixen l'estàndard IEEE-754 per a números normalitzats.

- **Output i postconditions:**

- **S:** producte dels factors A i B que segueix l'estàndard del IEEE-754 per a números normalitzats de 32 bits.

La seva estructura es troba descrita en la figura 5.4. Es pot observar que els operands que alimenten el circuit es descomponen segons les directives de l'estàndard. Cada camp és processat pel seu mòdul corresponent. Es pot observar com es compleixen les precondicions descrites anteriorment.

El seu camí crític és des del bit 2 de l'operand B ($B[2]$) fins al bit 2 del resultat ($S[2]$), causant un retard de 59,043 ns pel xip FPGA seleccionat. El seu consum és de 207,47 mW. Aquestes són les referències a optimitzar.

Cal destacar que el camí crític passa pel mòdul Wallace Tree de 24 bits, el qual és el més lent i el que ocupa més àrea de tot el disseny. Per aquesta raó, és lògic centrar les optimitzacions en aquest mòdul. Per una altra banda, no té sentit introduir error en segons que mòduls. En concret, el mòdul signe i normalitzador. El mòdul signe consta d'una porta lògica, fet que és irrellevant pel consum i retard del model. A més a més, un error de signe és fatal per la precisió de l'aproximació. En el cas del mòdul normalitzador, és necessari que es doni garantia que el resultat es normalitzi correctament i que es compleixin les postcondicions del mòdul.

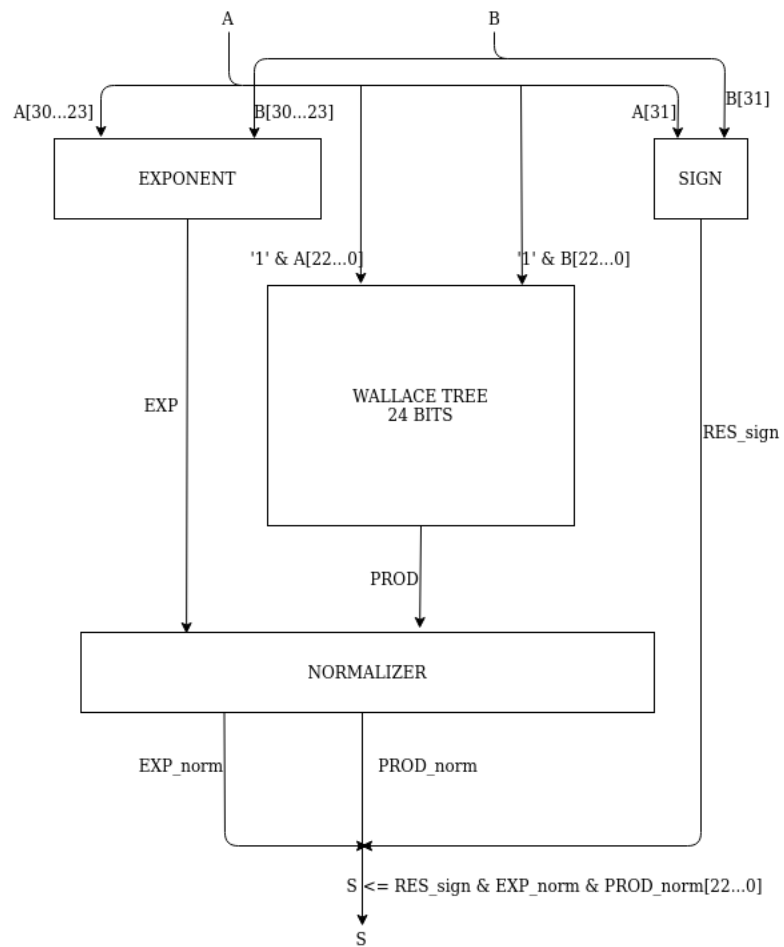


Figura 5.4: Diagrama del Multiplicador de Coma Flotant de 32 bits.
Elaboració pròpia.

Capítol 6

Models aproximats

En aquest capítol s'explica les aproximacions i models aproximats que s'han investigat en aquest treball. Diverses tècniques d'aproximació s'han estudiat. Amb cadascuna d'elles, s'han dissenyat models que parteixen del Model Base del capítol 5.

La majoria de les aproximacions aplicades són en el mòdul Wallace Tree de 24 bits descrit en el capítol 5.4. Segons la llei d'Amdahl [14], és més interessant enfocar les optimitzacions en els components que representin més retard. El multiplicador Wallace Tree, és la part que més retard provoca en el disseny. També és interessant estudiar com reduir el nombre de portes lògiques donat que és el mòdul que més àrea ocupa.

El multiplicador Wallace Tree compta amb un gran número de Full Adders i sumadors de propagació de carry (Ripple Carry Adder). Un factor que ajudarà a les aproximacions és l'estructura regular. Les aproximacions són especialment enfocades en els sumadors, els Full Adders i a estructura d'aquests.

6.1 Aproximacions mòdul Wallace Tree

Les aproximacions realitzades en el mòdul Wallace Tree de 24 bits s'han dividit en dos tipus:

- **Aproximacions en la suma de productes parcials:** els 3 sumadors de 48 bits són aproximats. Aquests sumaran de forma inexacta els 4 productes parcials generats pels mòduls Wallace Tree de 16 bits i 8 bits.
- **Aproximacions en la generació de productes parcials:** els dissenys que s'aproximen són els Wallace Tree de 8 i 16 bits que s'utilitzen per generar els productes parcials. En concret, les modificacions es poden trobar en tres localitzacions diferents:
 - **Wallace Tree 8 bits:** els sumadors de 16 bits que utilitzen aquest mòdul són els circuits que s'aproximen.
 - **Wallace Tree 16 bits:** els sumadors de 32 bits que utilitzen aquest mòdul són els circuits que s'aproximen.
 - **Wallace Tree 8 i 16 bits:** en aquest cas, els dos tipus sumadors anteriors s'aproximen al mateix temps.

6.2 Models Segmentats

La segmentació consisteix a dividir (segmentar) un procés en diverses etapes. Els processadors actuals executen instruccions segmentades. Les instruccions d'aquests processadors no s'executen completament en un cicle de rellotge, sinó que el seu camí de dades executa diverses etapes. Aquesta tècnica ha suposat un gran increment en el rendiment de la CPU, donat que permet executar diverses instruccions en diferents etapes alhora. Un altre avantatge que suposa, és la reducció dels camins de dades i en conseqüència del camí crític.

En aquest projecte no ens interessa incrementar el nombre d'operacions que es poden executar concurrentment en el mòdul multiplicador. La raó és que aquesta optimització no suposa cap aproximació. Per l'altra banda, dividir els camins de dades mitjançant la segmentació si que és una forma d'introduir error en els càlculs. Aquest és el concepte de segmentació que s'ha utilitzat en el projecte. Els models segmentats no asseguren la propagació del "carry" entre els bits de diferents posicions.

6.2.1 Sumadors segmentats

En els diferents models aproximats s'utilitzen una gran varietat de sumadors segmentats. Es poden distingir per dues característiques: nombre de bits (tamany dels operands) i nombre de seccions (cops que el sumador ha estat segmentat). Tots els sumadors parteixen de la mateixa base, el sumador RCA de la mateixa dimensió utilitzat en el model base.

El disseny és bastant simple. Consisteix en no connectar els bits "carry" dels operands consecutius que pertanyen a segmentacions diferents. Per tant, si un sumador és de "n" bits i té m segmentacions, el sumador presenta les següents característiques: Cada segmentació és de n/m bits i el "carry" es propaga des de la posició $(n/m)*k$ fins a la posició $(n/m)*k + ((n/m) - 1)$ on $m > k \geq 0$ i $k \in \mathbb{N}$.

El fet que el "carry" no es propagui per totes les posicions és el fet que introdueix error en el resultat de les sumes. En els dissenys se suposa que el "carry" d'entrada a un bloc segmentat és 0. Tenint en compte aquest fet, es poden fer les següents deduccions:

- **Resultat aproximat inferior:** el resultat dels sumadors segmentats és igual o inferior al sumador no segmentat quan no és te en compte el signe. El "carry" d'entrada entre blocs és sempre 0, quan existeix la possibilitat que la primera posició d'un bloc segmentat hauria de rebre un "carry" 1 del bloc anterior. Aquest fet fa que els resultats siguin iguals o inferiors al resultat exacte.
- **Relació entre nombre de segmentacions i error:** com s'ha mencionat en el punt anterior, l'error es genera en el moment que el bit "carry" que hauria d'alimentar el següent bloc segmentat té valor d'1. Per tant, com més blocs segmentats tingui el nostre disseny, més possibilitats que algun d'aquests "carrys" tingui valor 1.
- **Relació entre nombre de segmentacions i camí crític:** en un RCA, el camí crític recorre tots els generadors de "carry" de totes les posicions. Com a conseqüència, cada divisió d'aquest camí suposa una reducció del camí crític.

D'aquestes conclusions cal mencionar que el camí crític pot canviar si es creen massa blocs segmentats, per tant s'estaria introduint més error sense disminuir el retard del circuit.

Els models que s'han dissenyat seguint aquesta tècnica són els següents:

- Models que aproximen la suma de productes parcials:

- **Segmentat 2:** els sumadors de 48 bits s’han segmentat en 2 sumadors de mateixa mida.
- **Segmentat 4:** els sumadors de 48 bits s’han segmentat en 4 sumadors de mateixa mida.
- **Segmentat 8:** els sumadors de 48 bits s’han segmentat en 8 sumadors de mateixa mida.
- Models que aproximen la generació de productes parcials:
 - **SegHalf:** els sumadors de 16 i 32 bits s’han segmentat en 2 parts de mateixa mida.
 - **SegFour:** els sumadors de 16 i 32 bits s’han segmentat en 4 parts de mateixa mida.

6.3 Models especulatius

L’especulació es basa en predir el comportament. És una tècnica que també s’utilitza en processadors actuals. Un exemple són els “Branch Predictors” [3], aquests es basen en un circuit que la seva missió és esbrinar quina branca d’execució és la més probable. Una especulació similar és la que s’ha aplicat en els sumadors segmentats descrits anteriorment.

6.3.1 Sumadors especulatius

En els sumadors especulatius s’ha agafat com a base els sumadors segmentats i s’ha afegit un mòdul especulatiu. Aquest mòdul especulatiu fa un càlcul del qual podria ser el “carry” que ha d’alimentar un mòdul en concret. Això, tenint en compte els dos bits de posicions anteriors dels dos operands. En la següent figura trobem l’estructura del mòdul especulatiu.

Aquest mòdul ha estat inspirat en [22]. Tot i això, es podria considerar altres combinacions lògiques per especular sobre el carry o augmentar/disminuir el nombre de bits a tenir en compte.

Les característiques d’aquest mòdul especulatiu són:

- **Inputs i preconditions:**
 - **Amsb:** 2 bits de l’operand A que corresponen als bits de l’operand A en posicions de més pes d’un bloc sumador segmentat concret.
 - **Bmsb:** 2 bits de l’operand B que corresponen als bits de l’operand B en posicions de més pes d’un bloc sumador segmentat concret.
 - Els bits Amsb i Bmsb corresponen a bits del mateix pes dels operands A i B respectivament.
- **Outputs i postconditions:**
 - **Cso:** bit carry especulat per la posició següent.

Els models que s’han dissenyat seguint aquesta tècnica són els següents:

- Models que aproximen la suma de productes parcials:
 - **Especulatiu 2:** els sumadors de 48 bits s’han segmentat en 2 sumadors de mateixa mida els quals cadascun compta amb un modul especulador que alimenta el “carry” d’entrada.
 - **Especulatiu 4:** els sumadors de 48 bits s’han segmentat en 4 sumadors de mateixa mida.

- **Especulatiu 8**: els sumadors de 48 bits s'han segmentat en 8 sumadors de mateixa mida.

- Models que aproximen la generació de productes parcials:

- **SegHalf**: els sumadors de 16 i 32 bits s'han segmentat en 2 parts de mateixa mida.

- **SegFour**: els sumadors de 16 i 32 bits s'han segmentat en 4 parts de mateixa mida.

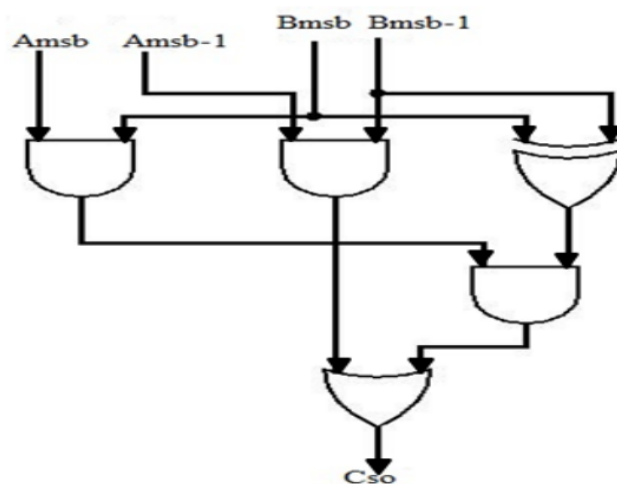


Figura 6.1: Mòdul Especulador.
Diagrama lògic del mòdul Especulador[22].

Amsb	Bmsb	Cso
00	00	0
00	01	0
00	10	0
00	11	0
01	00	0
01	01	1
01	10	0
01	11	1
10	00	0
10	01	0
10	10	0
10	11	0
11	00	0
11	01	1
11	10	0
11	11	1

Taula 6.1: Taula de veritat del mòdul Especulador.

Elaboació pròpia.

6.3.2 Sumadors especulatius amb compensador

Els sumadors especulatius generen “carry” segons les prediccions d’uns pocs bits anteriors. Aquesta predicció s’ha de realitzar abans que el mateix circuit sumador calculi el carry que ha estat especulat. D’aquesta forma, tots els blocs sumadors treballen alhora i el seu resultat és vàlid al mateix temps. Un cop el sumador especulatiu ha calculat el resultat, es pot saber si el “carry” especulat coincideix amb el que ha generat el bloc anterior.

La possibilitat de saber si s’ha especulat erròniament sense afegir circuit extra és característica única d’aquests sumadors entre tots els dissenys del projecte. Aquesta és la situació perfecta per integrar un mètode de control d’errors. En [22] es descriu un mètode per compensar possibles errors en la predicció.

Un nou mòdul que en aquest projecte s’ha anomenat Compensador es conté la lògica de la figura 6.2. Les característiques del circuit són:

- **Inputs i preconditions:**

- **Cso:** 1 bit que correspon al bit “carry” especulat.
- **Ci:** 1 bit que representa el bit “carry” generat pel bloc sumador anterior al bloc que ha especulat Cso.
- **Smsb:** 2 bits que corresponen a les dues posicions de més pes d’un sumador segmentat. S’han utilitzat per calcular Cso en el mòdul especulador.

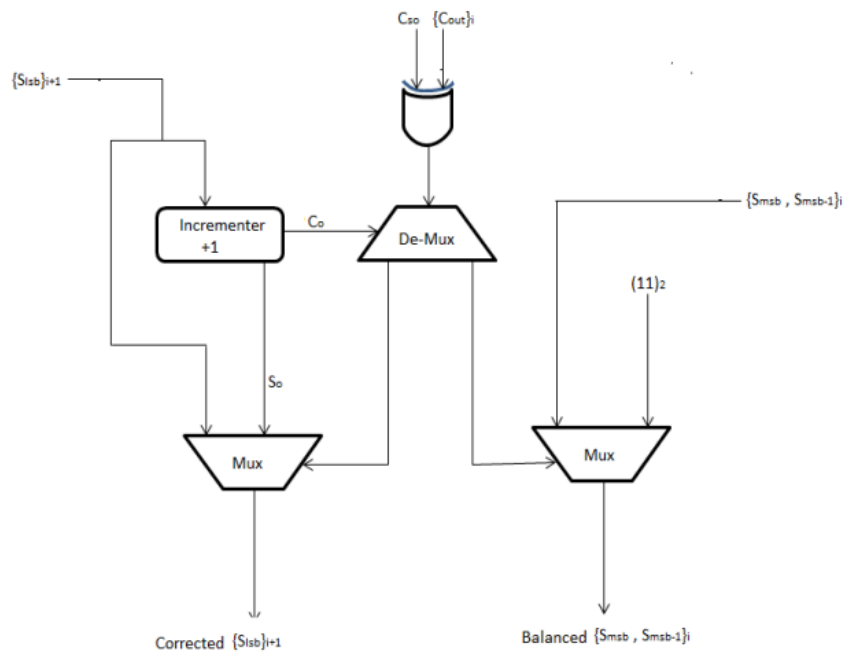


Figura 6.2: Mòdul Compensador.
 Diagrama lògic del mòdul Compensador[22].

- **Slsb**: 2 bits que corresponen a les dues posicions de menys pes del sumador segmentat següent.

- **Outputs i postconditions:**

- **Smsb_out**: 2 bits que són el resultat de compensar l'input Smsb.
- **Slsb_out**: 2 bits que són el resultat de compensar l'input Smsb.

En el cas que el “carry” especulat i el “carry” generat siguin diferents, s’ha de compensar l’error.

El compensador suposa que l’error és un “carry” que no s’ha predit. Per aquesta raó, el circuit intenta augmentar el resultat. S’ha de tenir en compte que si se suma 1 als bits Slsb, és possible que facin Overflow i el resultat sigui contraproduent. Per aquesta raó, quan Slsb val “11”, s’intenta compensar sonant el valor “11” als bits anteriors, als Slsb_out.

6.4 Sumador Low OR

Els sumadors Low OR són un altre tipus de sumador que es caracteritzen per la substitució dels Full Adders de posicions baixes per portes lògiques OR. D’aquesta manera es redueix considerablement la quantitat de circuits i les seves connexions.

S’han creat diversos sumadors Low OR. Es caracteritzen segons:

- **Nombre de bits**: quantitat de bits del sumador.
- **Bits Low OR**: quantitat de Full Adders que s’han substituït per portes OR.

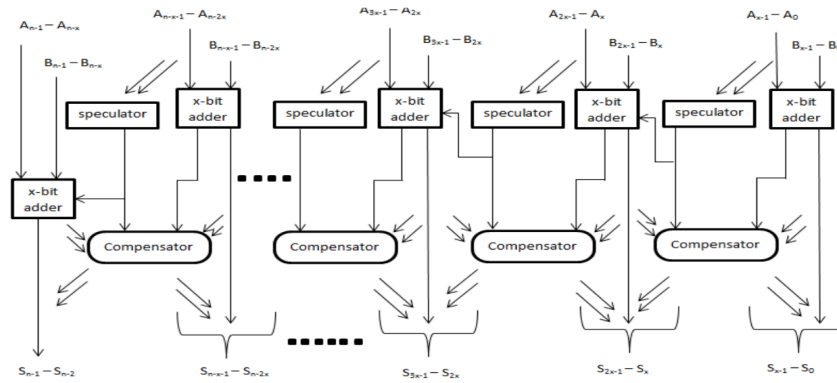


Figura 6.3: Sumador Especulatiu amb Compensadors.
 Diagrama d'un sumador especulatiu amb compensadors d'errors[22].

Tenint en compte aquests paràmetres, s'han dissenyat diversos sumadors de diferents mides i amb diferents números de portes lògiques OR. En concret, aquests són els dissenys utilitzats en l'estudi:

- Sumador de 48 bits, utilitzat per l'aproximació de la suma de productes parcials:
 - 8 portes lògiques OR.
 - 16 portes lògiques OR.
 - 24 portes lògiques OR.
 - 32 portes lògiques OR.
- Sumador de 32 bits, utilitzat per l'aproximació de la generació de productes parcials:
 - 8 portes lògiques OR.
 - 16 portes lògiques OR.
- Sumador de 16 bits, utilitzat per l'aproximació de la generació de productes parcials:
 - 4 portes lògiques OR.
 - 8 portes lògiques OR.

6.5 Full Adders Aproximats

L'estructura del multiplicador de coma flotant utilitza una gran quantitat de sumadors Full adders. És interessant pel projecte estudiar diferents alternatives de Full Adders. En [21], es presenten 12 tipus diferents de Full Adders inexacts (també referenciats durant el projecte com Approximate Adders o AA). En l'article, també s'estudia l'efecte d'aquests nous circuits en l'un multiplicador Dadda Tree.

En la següent taula trobem la lògica que utilitza aquests circuits:

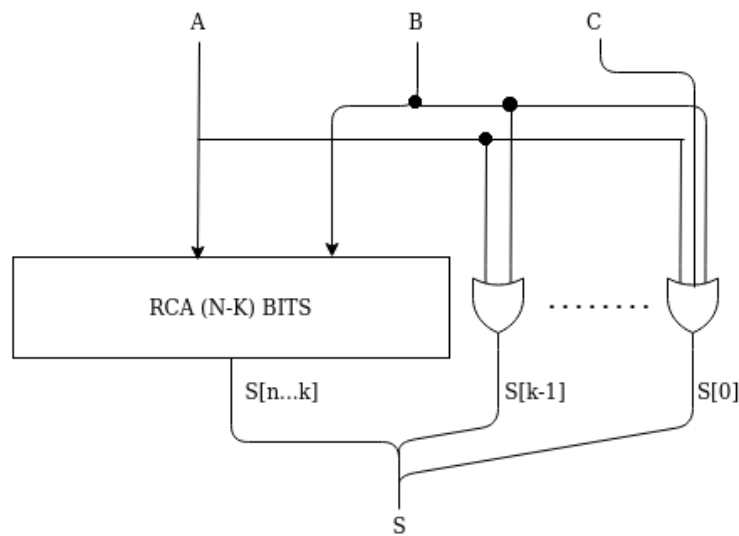


Figura 6.4: Sumador Low OR

Diagrama de l'Arquitectura del sumador Low OR, on "n" representa el número de bits del sumador i "k" el número de portes OR utilitzades.

Full Adder	SUM	C OUT
AA1	$ABC_{in} + C_{out}'C_{in}$	$AC_{in} + B$
AA2	C_{out}'	$AB + BC_{in} + AC_{in}$
AA3	C_{out}'	$AC_{in} + B$
AA4	$ABC_{in} + C_{out}'C_{in}$	A
AA5	B	A
AA6	$A' + BC_{in}$	A
AA7	$A'(B+C_{in}) + BC_{in}$	$AB + BC_{in} + AC_{in}$
AA8	$(A'+B)C_{in}$	$AB + BC_{in} + AC_{in}$
AA9	$A'B' + B'C_{in}' + ABC_{in} + A'BC_{in}'$	$AB + BC_{in} + AC_{in}$
AA10	$A' + BC_{in}$	$AB + BC_{in} + AC_{in}$
AA11	$A'(B'+C_{in}') + A'BC_{in}'$	$AB + BC_{in} + AC_{in}$
AA12	$AB + BC_{in}' + A'B'C_{in} + AB'C_{in}'$	$AB + BC_{in} + AC_{in}$

Taula 6.2: Lògica Approximate Adders.

Lògica dels components Approximate Adders. A i B denoten els operands d'entrada i C_{in} denota el carry d'entrada. L'operació AND entre les entrades A i B s'indica com AB, l'operació OR com

$A + B$ i l'operació NOT com A' .

Truth table for existing AAs (AA1-AA5).

Inputs			Outputs											
A	B	C _{in}	FA		AA1		AA2		AA3		AA4		AA5	
			Sum	C _{out}	Sum	C _{out}	Sum	C _{out}	Sum	C _{out}	Sum	C _{out}	Sum	C _{out}
0	0	0	0	0	0	0	1	0	1x	0	0	0	0	0
0	0	1	1	0	1	0	1	0	1	0	1	0	0x	0
0	1	0	1	0	0x	1x	1	0	0x	1x	0x	0	1	0
0	1	1	0	1	0	1	0	1	0	1	1x	0x	1x	0x
1	0	0	1	0	0x	0	1	0	1	0	0x	1x	0x	1x
1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
1	1	0	0	1	0	1	0	1	0	1	0	1	1x	1
1	1	1	1	1	1	1	0x	1	0	1	1	1	1	1

Note: x denotes incorrect output.

Figura 6.5: Taula de veritat AA[1...5]
Taula de veritat dels Full Adders aproximats del 1 al 5[21] .

Truth table for proposed AAs (AA6-AA12).

Inputs			Outputs															
A	B	C _{in}	FA		AA6		AA7		AA8		AA9		AA10		AA11		AA12	
			Sum	C _{out}	Sum	C _{out}	Sum	C _{out}	Sum	C _{out}	Sum	C _{out}	Sum	C _{out}	Sum	C _{out}	Sum	C _{out}
0	0	0	0	0	1x	0	0	0	0	1x	0	1x	0	1x	0	0	0	0
0	0	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
0	1	0	1	0	1	0	1	0	0x	0	1	0	1	0	1	0	1	0
0	1	1	0	1	1x	0x	1x	1	1x	1	0	1	1x	1	0	1	0	1
1	0	0	1	0	0x	1x	0x	0	0x	0	1	0	0x	0	1	0	1	0
1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
1	1	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1x	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0x	1	1	1

Note: x denotes incorrect output.

Figura 6.6: Taula de veritat AA[6...12]
Taula de veritat dels Full Adders aproximats del 6 al 12[21] .

Aquests 12 Full Adders aproximats incrementen considerablement el nombre de models que es poden dissenyar. És interessant també investigar quina és la lògica que més profit donen al multiplicador de coma flotant. En aquest estudi, per cada tipus d'optimització, es dissenyarà un model amb cada Full Adder. En concret tenim aquests dissenys:

- Optimització mitjançant l'aproximació de la **suma de productes parcials**:
 - **Sumador de 48 bits**: Ripple Carry Adder que substitueix tots els Full Adders per Approximate Adders. 12 models diferents en total. Es referència aquests models com WallaceAA_n, on n és un número de l'1 al 12 que indica el AA que utilitza.
- Optimització mitjançant l'aproximació de la **generació de productes parcials**: L'estructura dels mòduls recursius del Wallace Tree s'han modificat Les estructures del Wallace Tree de 16 i de 8 bits. Els productes parcials s'han modificat de forma similar al punt anterior. S'ha realitzat aquest mòdul:
 - **Wallace025AA_n**: els sumadors de 16 bits i 32 bits hi han substituït el 25% de Full Adders la part baixa per Approximate Adders. El paràmetre "n" és un número de l'1 al 12 que indica el AA utilitzat en el disseny.
 - **Wallace050AA_n**: els sumadors de 16 bits i 32 bits hi ha substituït el 50% de Full Adders la part baixa per Approximate Adders. El paràmetre "n" és un número de l'1 al 12 que indica el AA utilitzat en el disseny.

En total es compti en el projecte amb 36 models aproximats on s'han utilitzat exclusivament Full Adders aproximats en el mòdul Wallace Tree.

6.6 Aproximacions en el mòdul Exponent

Les aproximacions anteriors s'han centrat en el mòdul Wallace Tree. En el projecte també s'han dissenyat models que no aproximem el multiplicador de mantisses. La importància que han rebut aquestes modificacions no és la mateixa. En concret, s'ha aplicat les tècniques de segmentació, especulació i especulació amb compensador en el mòdul exponent. El resultat són tres models nous que seran d'utilitat per estudiar els efectes d'aproximacions en l'exponent. Els dissenys són els següents:

- **expoSeg**: model creat segmentant el sumador RCA de 8 bits del mòdul exponent en dos sumadors de la mateixa mida.
- **expoSpec**: model creat segmentant el sumador RCA de 8 bits del mòdul exponent en dos sumadors de la mateixa mida. Cada sumador és alimentat per un mòdul especulador en el "carry" d'entrada.
- **expoComp**: expoSpec :model creat segmentant el sumador RCA de 8 bits del mòdul exponent en dos sumadors de la mateixa mida. Cada sumador és alimentat per un mòdul especulador en el "carry" d'entrada i amb un compensador entre els dos sumadors.

No s'han dissenyat models que utilitzin Approximate Adders pel fet que els models d'escrits ja s'ha considerat suficients per estudiar l'efecte d'aproximacions en l'exponent.

6.7 Aproximacions en el mòdul Normalitzador

L'últim model presentat en el projecte és l'únic que inclou modificacions en el mòdul Normalitzador. En la secció 5.5, s'ha comentat que aquest mòdul és un disseny optimitzat pensat per funcionar en la normalització de productes de nombre normals. Per tant, els resultats no són tan transportables a productes comercials com ho poden ser les modificacions als mòduls Wallace Tree i Exponent, però s'ha considerat interessant estudiar l'efecte de les modificacions en els errors produïts.

Normal OR és el nom que s'ha donat al model encarregat de presentar-nos aquesta aproximació. La modificació que presenta és la substitució del sumador que era l'encarregat de sumar 1 a l'exponent per una simple porta OR en el bit més baix de l'exponent. Aquest bit de menys pes haurà de combinar-se en la porta OR amb el senyal que indica si ha fet falta desplaçar la mantissa. D'aquesta forma, s'ajusta l'exponent en cas de desplaçament de mantissa d'una forma que substituïm un sumador RCA per una versió d'una única porta del Sumador Low OR.

Capítol 7

Anàlisi

Les dades extretes en les simulacions s'han d'estudiar com a fi de determinar els punts forts i dèbils de cada circuit. En aquesta secció, es fa l'anàlisi dels resultats obtinguts. El criteri utilitzat no és directe, perquè aquí es presenten mètriques que suposen millora, com la reducció del retard del model o reducció del consum, i altres mètriques negatives, com els errors. Donat aquest cas, és important establir un raonament per distingir quines optimitzacions han estat més efectives. S'ha de tindre en compte que errors amb grans magnituds poden eclipsar qualsevol millora de rendiment i que aquesta optimització ja no sigui justificable. Un altre criteri a considerar és la pròpia distribució de l'error. Per aquesta raó, els quartils, secció 4.4, s'hi han inclòs per obtenir més informació del comportament del model. Durant el capítol hi ha diverses taules que mostren les següents mètriques sobre els models:

- **Ret(ns)**: retard del circuit en nanosegons.
- **Con(mW)**: consum energètic del circuit en miliWatts.
- **Md.E(%)**: mediana d'error relatiu en percentatge.
- **E.mx.(%)**: percentatge d'error màxim produït pel model en les simulacions.
- **Sp.UP(%)**: percentatge de Speed Up del model aproximat respecte al model base.
- **Est.OP(%)**: percentatge d'estalvi energètic per operació que produeix el model aproximat respecte al model base.

Els models s'han classificat en diferents tipus, tal com s'explica en la secció 6. Per tant, en l'anàlisi no pot faltar un estudi de com els diferents models de cada tipus es comporta. A més a més, es compara els resultats entre tipus per obtenir una visió més àmplia de les optimitzacions. Segons la convenient de l'anàlisi, s'han escollit alguns models per fer les comparatives.

7.1 Resultats aproximació Wallace Tree

Les aproximacions en el Wallace Tree suposen la majoria dels dissenys que s'han realitzat en el projecte. En aquesta secció, es veu quins són els resultats que aquests models han obtingut.

7.1.1 Aproximació en la suma de productes parcials

Low OR

Els sumadors Low OR són les aproximacions més senzilles que s'ha utilitzat en el projecte. En aquest cas, s'han utilitzat tres sumadors de 48 bits Low OR per a cada un dels models aproximats. La descripció d'aquest model es troba en la secció 6.4.

Els models ADD OR 8 i ADD OR 16 no han presentat error respecte al model base en cap de les 150.000 operacions. Per contra, tampoc s'ha obtingut cap reducció de retard ni consum. És clar de concloure que aquestes millores no tenen un impacte suficient en els resultats ni tampoc en la síntesi del circuit per com distingir-se del model base.

Model	Ret(ns)	Con(mW)	Md.E.(%)	E.mx.(%)	Sp.UP(%)	Est.OP(%)
Base	59,043	207,47	-	-	-	-
ADD OR 8	59,043	207,47	0,0	0,0	0,0	0,0
ADD OR 16	59,043	207,47	0,0	0,0	0,0	0,0
ADD OR 24	50,129	202,25	0,000009463	0,00004643	17,78	17,23
ADD OR 32	44,905	190,94	0,0010379	0,0089307	31,48	30,00

Taula 7.1: Mètriques models Low OR.

Mètriques de rendiment i error dels models Low Or que aproximen la suma de productes parcials del mòdul Wallace Tree de 24 bits.

Aquest no és el cas dels models ADD OR 24 i ADD OR 32, els quals si presenten errors i optimitzacions en les simulacions realitzades. Com a punt a favor, els errors màxims són de magnituds molt petites; 0,000046% per ADD OR 24 i 0,008930% per ADD OR 32. Tenint en compte que aquest és el percentatge d'error màxim durant el test, podem concloure que es tracta d'un error que no distorsiona considerablement el resultat. Els retards dels models s'han reduït fins a 50,129 ns pel model ADD OR 24 i fins 44,905 ns per ADD OR 32. El consum d'aquests dos models també s'ha reduït, a 202,25 mW i 190,94 mW respectivament. Si la reducció de temps crític dels models es combina amb l'estalvi energètic, tenim que els models necessiten un 17,2% i un 30% menys d'energia per calcular una operació.

Segmentats

Els sumadors segmentats són l'arquitectura més senzilla que es basa en la manipulació del "carry" en els sumadors RCA d'aquest projecte. Segons el grau de segmentació, nombre de cops que s'ha segmentat cada sumador, es veu increment d'error i grau d'optimització. Destacar que les mètriques obtingudes són satisfactòries. L'error es manté en cotes baixes, Segmentat 8 és el model que més error presenta. Encara així, el seu quartil 0.75 es manté a 0.08% d'error relatiu. Per tant, és un error que podria ser acceptable per una unitat funcional aproximada. Per l'altra banda, s'ha de tindre en compte l'error màxim d'aquest model que s'acosta fins al 6% en les simulacions. El model Segmentat 4 es manté en un ordre d'error similar a Segmentat 8, però amb un error màxim molt més petit, d'un 0.18%. Aquest fet es pot atribuir a l'error en casos concrets on el "carry" no es propaga en posicions altes. El model Segmentat 2, en una altra banda, presenta uns errors tan petits respecte ala seva variant que l'orde de magnitud és completament diferent. En les simulacions s'han extret un error inferior al 0,0000067% en el 50% de les operacions i un error

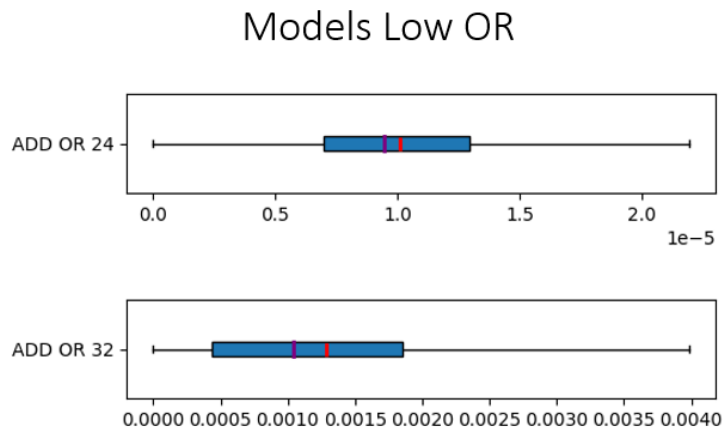


Figura 7.1: Boxplot Models Low OR

Boxplot on es mostra la distribució d'error dels models ADD OR 24 i ADD OR 32. Línia púrpura correspon a la mediana i la línia vermella a la mitja. Elaboració pròpia.

màxim de 0,000023%.

Model	Ret(ns)	Con(mW)	Md.E.(%)	E.mx.(%)	Sp.UP(%)	Est.OP(%)
Base	59,043	207,47	-	-	-	-
Segmentat 2	53,595	204,32	0,0000067246	0,000023218	10,17	10,61
Segmentat 4	44,581	194,4	0,042318	0,187625	32,44	29,25
Segmentat 8	37,474	188,59	0,048162	6,053475	57,56	42,31

Taula 7.2: Mètriques models Segmentats.

Mètriques de rendiment i error dels models Segmentat que aproximen la suma de productes parcials del mòdul Wallace Tree de 24 bits.

Els models Segmentats en aquesta aproximació de la suma de productes parcials també presenta una gran millora de consum i retard. Segons el grau de segmentació, més gran és l'optimització. De base, el model Segmentat 2 presenta bones mètriques d'optimització; retard de 53,595 ns i una reducció del 10,6% d'energia per operació. Tenint en compte el seu baix error, es pot considerar una optimització interessant. Tenint en compte l'error, ja s'ha mencionat que Segmentat 4 i Segmentat 8 tenen comportament similar, també ho fan respecte a les optimitzacions. Els seus retards són de 44,581 nanosegons i 37,474 nanosegons respectivament. Aquestes reduccions del camí crític sumat a les reduccions de consum generen un estalvi energètic del 29,25% per Segmentat 4 i de 42,3% per Segmenat 8.

Aquests tres models formen un conjunt d'alternatives que un possible arquitecte de computadors podria escollir segons l'estalvi energètic que necessita i l'error que les seves aplicacions poden suportar.

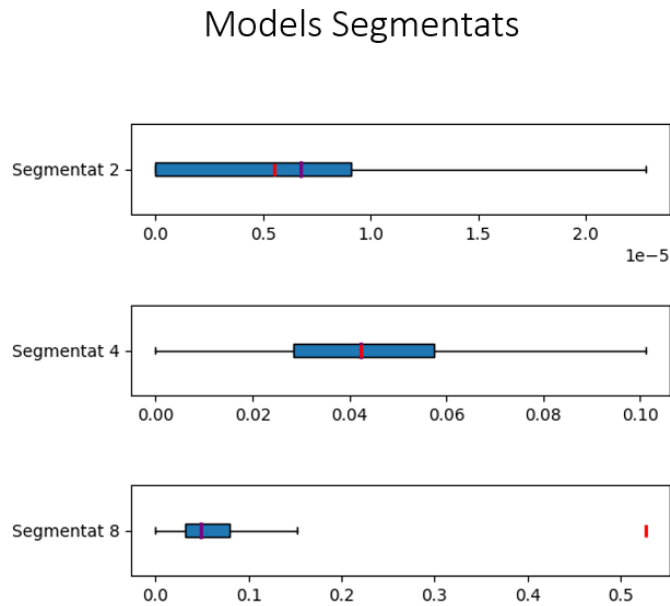


Figura 7.2: Boxplot Models Segmentats

Boxplot on es mostra la distribució d'error dels models Segmentat 2, Segmentat 4 i Segmentat 8. Línia púrpura correspon a la mediana i la línia vermella a la mitja. Elaboració pròpia.

Especulatius

Els models Especulatius han obtingut mètriques interessants. Donat que en aquest cas s'ha afegit el mòdul especulador als models segmentats, l'error s'ha vist reduït considerablement. Pel model Especulatiu 2, més de la meitat d'operacions s'efectuen sense errors respecte al model Base. Aquesta reducció d'error també és notable en Especulatiu 4, que ha passat del 0,04% d'error mitjà de Segmentat 4 a un 0,000011%. Especulatiu 8 també ha aconseguit reduir els errors dels seus quartils, però no de la mateixa manera amb l'error màxim que continua a 6,02%.

La reducció d'aquest error també ha implicat una reducció en el grau d'optimitzacions. Especulatiu 2 presenta un retard de 59,444 ns, superior al model Base. Encara així, el seu consum ha baixat un 12% donat que els tres models necessiten aproximadament uns 30 mW menys per operar. Especulatiu 4 ha aconseguit baixar el seu retard fins al 54,651 ns i Especulatiu 8 fins a 50,127 ns. El consum per operació s'ha reduït en 19,27% i 26,42%.

Model	Ret(ns)	Con(mW)	Md.E.(%)	E.mx.(%)	Sp.UP(%)	Est.OP(%)
Base	59,043	207,47	-	-	-	-
Especulatiu 2	59,444	181,12	0,0000067246	0,00004568	-0,67	12,11
Especulatiu 4	54,651	180,94	0,042318	0,187601	8,04	19,27
Especulatiu 8	50,127	179,8	0,048162	6,026826	17,79	26,42

Taula 7.3: Mètriques models Especulatiu.

Mètriques de rendiment i error dels models Especulatiu que aproximen la suma de productes parcials del mòdul Wallace Tree de 24 bits.

Models Especulatius

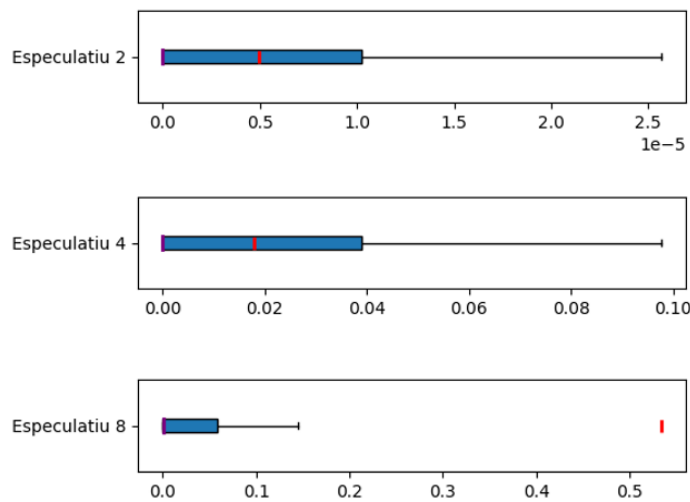


Figura 7.3: Boxplot Models Especulatius

Boxplot on es mostra la distribució d'error dels models Especulatiu 2, Especulatiu 4 i Especulatiu 8. Línia púrpura correspon a la mediana i la línia vermella a la mitja. Elaboració pròpia.

Compensadors

Els models Compensadors han donat el resultat esperat. L'error que presenten cada model Especulatiu s'ha vist reduït pel sistema de control d'errors implementat. En el cas de Compensador 2, s'ha aconseguit reduir el Quartil 0,75 a un 0% d'error. Pels models Compensador 4 i Compensador 8 també presenten errors inferiors als models similars sense compensador. Un fet important és la reducció de l'error màxim en el cas del Compensador 8 a un 2,09%. Aquest error no s'havia aconseguit reduir en aquesta proporció amb el salt de model Segmentat a model Especulatiu.

Afegir aquest sistema de correcció d'errors ha afectat negativament a les optimitzacions que presenten els models anteriors. Primerament, Compensador 2 veu augmentat el seu retard de

forma considerable, fins als 63,923 ns. Respecte al model Base suposa un increment de latència que possiblement no és desitjable. El seu consum es veu reduït un 12%, una reducció lleugerament superior al seu germà Especulatiu 2. Per continuar, els models Compensador 4 i Compensador 8 sí que presenten un speed up de 7,26% i 9,95%.

Model	Ret(ns)	Con(mW)	Md.E.(%)	E.mx.(%)	Sp.UP(%)	Est.OP(%)
Base	59,043	207,47	-	-	-	-
Compensador 2	63,923	180,23	0,0	0,000046401	-7,63	5,95
Compensador 4	55,043	188,4	0,00001170	0,192898	7,27	15,34
Compensador 8	53,696	184,38	0,000006391	2,09125	9,96	19,18

Taula 7.4: Mètriques models Compensador.

Mètriques de rendiment i error dels models Compensador que aproximen la suma de productes parcials del mòdul Wallace Tree de 24 bits.

Models Compensador

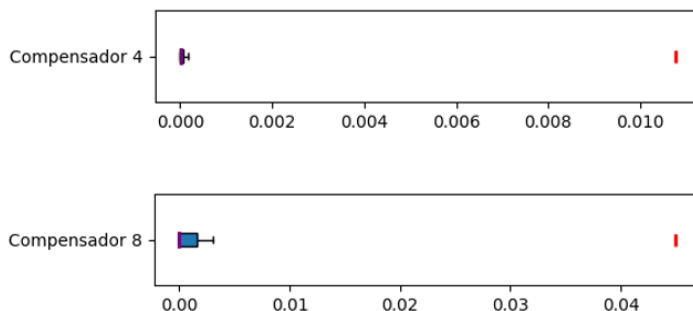


Figura 7.4: Boxplot Models Compensador

Boxplot on es mostra la distribució d'error dels models Compensador 4 i Compensador 8. Línia púrpura correspon a la mediana i la línia vermella a la mitja. Elaboració pròpia.

Full Adders aproximats

Els models amb Approximate Adders segueixen una línia diferent dels models anteriors. Els circuits presenten optimitzacions bones, entre el 10% (AA2) i el 38,88% (AA6) de speed up. En contra, els errors són excessivament grans; els Quartils 0,50 es troben majoritàriament al voltant del 54% d'error. Un error que distorsiona qualsevol resultat i amb el qual difícilment es podria fer funcionar cap algoritme.

Aquest no és el cas dels models AA6 i AA12. AA6 té una mediana inferior, el 10%, però la seva mitjana puja fins al 24% donat que els sí que compta en casos on l'error es dispara. El seu error màxim arriba fins al 99,74%, superior als models on la mediana era del 54%. Per tant, es pot concloure que el model AA6 té un error menys estable. El cas contrari és el model AA12. Aquest

ha obtingut l'error més petit de tots els models que aproximen la suma de productes parcials amb Approximate Adders. La seva mitjana és del 0,14% i té un error màxim de l'1,88%. Per l'altra banda, el model AA12 presenta un speed up negatiu, del -3,18% i un estalvi energètic de només l'1,7%.

Model	Ret(ns)	Con(mW)	Md.E.(%)	E.mx.(%)	Sp.UP(%)	Est.OP(%)
Base	59,043	207,47	-	-	-	-
AA1	47,045	178,32	54,04911	74,96648	25,50	31,52
AA2	53,258	167,02	54,04911	74,96965	10,86	27,38
AA3	47,111	197,00	54,06146	74,96960	25,33	24,24
AA4	37,674	165,36	32,75024	74,96354	56,72	49,14
AA5	34,025	174,96	54,06695	74,96858	73,53	51,40
AA6	42,511	177,6	10,02744	99,74324	38,89	38,37
AA7	45,638	188,7	54,05629	74,96708	29,37	29,70
AA8	44,528	179,05	54,06332	74,96965	32,60	34,91
AA9	58,250	187,91	82,28430	298,61907	1,36	10,64
AA10	42,585	185,34	82,35498	298,59889	38,68	35,57
AA11	49,825	163,18	53,07470	74,96965	18,50	33,63
AA12	60,981	197,41	0,09074	1,33869	-3,18	1,73

Taula 7.5: Mètriques models Approximate Adders.

Mètriques de rendiment i error dels models Approximate Adders que aproximen la suma de productes parcials del mòdul Wallace Tree de 24 bits.

7.1.2 Aproximació en la generació de productes parcials

Els models que aproximen la generació de productes parcials tenen un comportament diferent en terme d'errors i de consums al comportament que ho fan els models que aproximen la suma de productes parcials. En aquesta secció s'analitza els resultats dels 34 models els quals han aplicat les tècniques d'aproximacions.

Low OR Wallace Tree

Els sumadors Low Or s'han utilitzat per substituir els sumadors RCA que generen els productes parcials. Els models SUM 16 OR 4 i SUM 32 OR 16 es comporten de forma similar a com ho fa els models ADD OR 8 i ADD OR 16. Aquests dos nous models tampoc aporten un alt grau d'aproximacions, per tant els quartils es troben a un 0% d'error, igual que l'error màxim de la simulació. Per l'altra banda, les eines d'anàlisi no han estimat ni estalvi energètic ni tampoc millora en el temps de càlcul.

Els models SUM 16 OR 8 i SUM 32 OR 16 sí que presenten error. Les seves medianes són de 0,0028% i 0,00034% respectivament. Els seus errors màxims també són petits, de l'1,67% i 0,005%. Tenint en compte aquestes dades, es pot establir que aquests circuits presenten cotes d'error amb

Approximate Adders

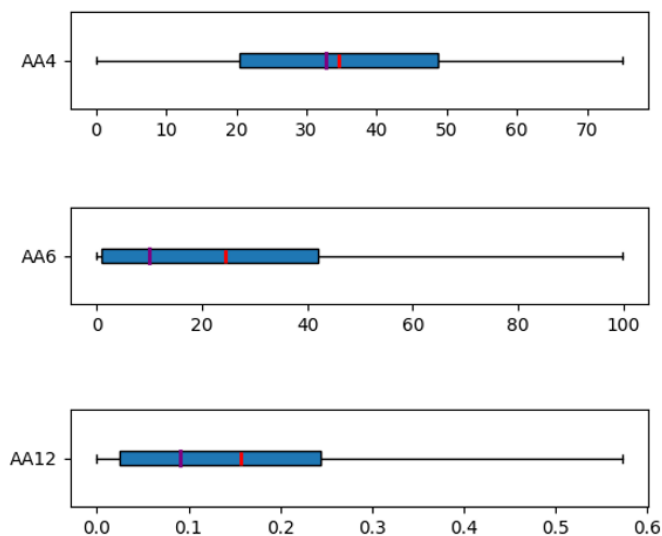


Figura 7.5: Boxplot Models AA

Boxplot on es mostra la distribució d'error dels models AA4, AA6 i AA12. Línia púrpura correspon a la mediana i la línia vermella a la mitja. Elaboració pròpia.

les quals alguns algoritmes podrien treballar. El model SUM 16 OR 8 té un speed up de 102,29% i un estalvi energètic del 7,45%, SUM 32 OR 16 té un speed up de 18,33% i un estalvi energètic de 20,76%.

Model	Ret(ns)	Con(mW)	Md.E.(%)	E.mx.(%)	Sp.UP(%)	Est.OP(%)
Base	59,043	207,47	-	-	-	-
SUM 16 OR 4	59,043	207,47	0,0	0,0	0,0	0,0
SUM 16 OR 8	57,720	196,42	0,16649	1,66561	2,29	7,45
SUM 32 OR 8	59,043	207,47	0,0	0,0	0,0	0,0
SUM 32 OR 16	49,895	194,54	0,00087753	0,0050275	18,33	20,76

Taula 7.6: Mètriques models Low OR Wallace Tree.

Mètriques de rendiment i error dels models Low OR que aproximen la generació de productes parcials del mòdul Wallace Tree de 24 bits.

El model SUM 32 OR 16 produeix menys error en les operacions i té millor mètriques d'optimització. Es pot observar que en aquest cas, l'aproximació que s'hi ha realitza en un nivell més proper de la generació final dels productes parcials, han obtingut millor resultats.

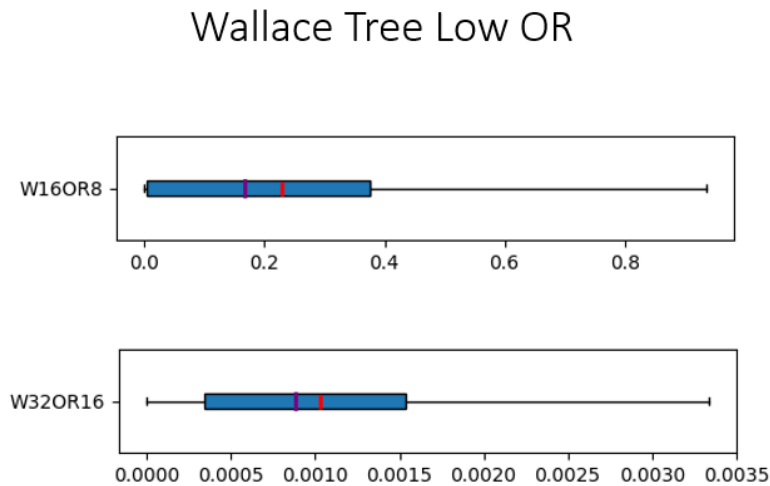


Figura 7.6: Boxplot Models Wallace Tree Low OR.

Boxplot on es mostra la distribució d'error dels models W16OR8 i W32OR16. Línia púrpura correspon a la mediana i la línia vermella a la mitja. Elaboració pròpia.

Segmentats Wallace Tree

A continuació s'analitza els resultats de les simulacions on els productes parcials han estat calculats amb sumadors RCA segmentats. S'han dissenyat dos models: SegHalf i SegFour. SegHalf s'ha dissenyat segmentant en dues parts de la mateixa grandària els sumadors de 16 bits i 32 bits que utilitzen els mòduls Wallace Tree 8 i Wallace Tree 16 respectivament. SegFour s'ha dissenyat de forma similar, però els mateixos sumadors s'han segmentat equitativament en 4.

Model	Ret(ns)	Con(mW)	Md.E.(%)	E.mx.(%)	Sp.UP(%)	Est.OP(%)
Base	59,043	207,47	-	-	-	-
SegHalf	46,763	198,15	0,48777	2,29747	26,26	24,36
SegFour	50,464	186,7	8,11625	25,60327	17,00	23,09

Taula 7.7: Mètriques models Segmentats Wallace Tree.

Mètriques de rendiment i error dels models Segmentats que aproximen la generació de productes parcials del mòdul Wallace Tree de 24 bits.

Les mètriques d'error obtingudes en les simulacions ens mostra que el model SegFour calcula de forma molt més imprecisa que la seva versió amb menys segmentació, SegHalf. En concret, la mitjana d'error de SegFour és de 6,62% en contra del 0,42% de SegHalf. Una característica positiva de SegHalf és que manté acotat l'error dins d'uns valors acceptables, 2,30% d'error màxim en les simulacions, en contrast amb el 25,60% de SegFour.

Les millores són positives respecte al model Base, tots dos models redueixen retard i consum. El consum energètic de SegFour és més petit que el consum de SegHalf, 186,7 mW respecte a 198,15 mW. Però el fet que SegFour presenti un retard major, fa que l'estalvi d'energia per operació no sigui major.

Wallace Tree Segmentat

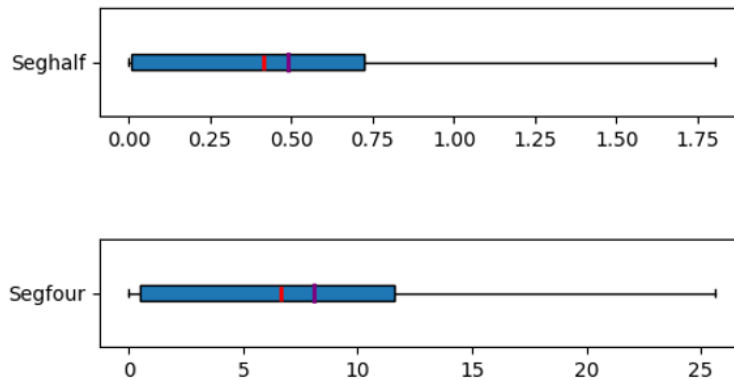


Figura 7.7: Boxplot Models Wallace Tree Segmentat
 Boxplot on es mostra la distribució d'error dels models Seghalf i Segfour. Línia púrpura correspon a la mediana i la línia vermella a la mitja. Elaboració pròpia.

En tot cas, el model SegHalf presenta característiques més interessants donat que té un rendiment òptim que el model base, 24,36% d'estalvi energètic per multiplicació i un speed up del 26,26%, i l'error introduït és controlable.

Espectatius Wallace Tree

Els models d'aquesta secció s'han dissenyat afegint el mòdul especulatiu al "carry" segmentat dels circuits SegHalf i SegFour, creant els models SpecHalf i SpecFour. Els nous dissenys han reduït la magnitud de l'error de les seves versions segmentades. Les mitjanes s'han reduït a 0,18% per SpecHalf i a 3,52% per SpecFour. Cal destacar el cas de l'error màxim de SpecFour, que gairebé s'ha triplicat fins a arribar al 73,26%. Aquest error fa que aquest model difícilment es pugui utilitzar en una aplicació real.

Model	Ret(ns)	Con(mW)	Md.E.(%)	E.mx.(%)	Sp.UP(%)	Est.OP(%)
Base	59,043	207,47	-	-	-	-
SpecHalf	58,523	175,35	0,0027613	2,23243	0,89	16,23
SpecFour	58,650	183,16	0,065823	73,26017	0,67	12,30

Taula 7.8: Mètriques models Espectatius Wallace Tree.

Mètriques de rendiment i error dels models Espectatius que aproximen la generació de productes parcials del mòdul Wallace Tree de 24 bits.

Les optimitzacions són discretes. En termes de millora de velocitat, no s'hi ha aconseguit reduir el retard dels models considerablement. El speed up obtingut ha estat de 100,89% per SpecHalf i

de 0,67% per SpecFour. El consum energètic sí que s'ha reduït de forma satisfactòria, estalviant un total del 16,23% al SpecHalf i del 12,30% al SpecFour d'estalvi d'energia.

Wallace Tree Especulatiu

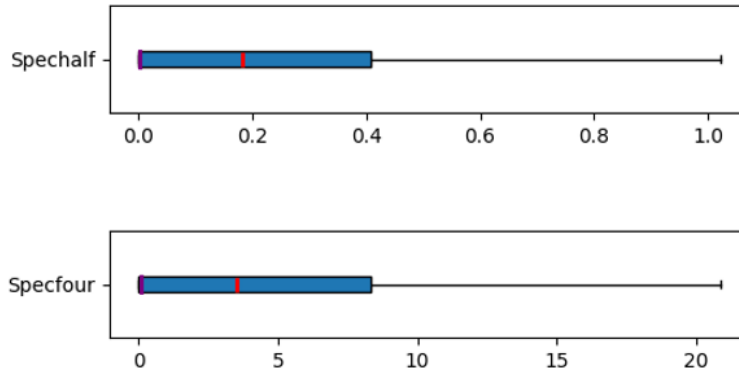


Figura 7.8: Boxplot Models Wallace Tree Especulatiu.

Boxplot on es mostra la distribució d'error dels models Spechalf i Specfour. Línia púrpura correspon a la mediana i la línia vermella a la mitja. Elaboració pròpia.

Compensadors Wallace Tree

Els models Compensadors han intentat corregir l'error d'especulació que han introduït els models SpecHalf i SpecFour en l'especulació de la generació de productes parcials. Per això, s'ha afegit mòduls compensadors en i s'han dissenyat els models CompHalf i CompFour. Desgraciadament, aquests models no han aconseguit reduir l'error de les seves versions no compensades i l'han superat lleugerament en les simulacions realitzades.

Model	Ret(ns)	Con(mW)	Md.E.(%)	E.mx.(%)	Sp.UP(%)	Est.OP(%)
Base	59,043	207,47	-	-	-	-
CompHalf	57,793	178,18	0,00045929	2,46703	2,16	15,94
CompFour	61,608	189,39	0,0096630	73,23892	-4,16	4,75

Taula 7.9: Mètriques models Compensador Wallace Tree.

Mètriques de rendiment i error dels models Compensador que aproximen la generació de productes parcials del mòdul Wallace Tree de 24 bits.

Les millores de rendiment només es presenten en el model CompHalf, que estalvia un 6,855% d'energia per operació i té un speed up de 2,16%.

Wallace Tree Compensador

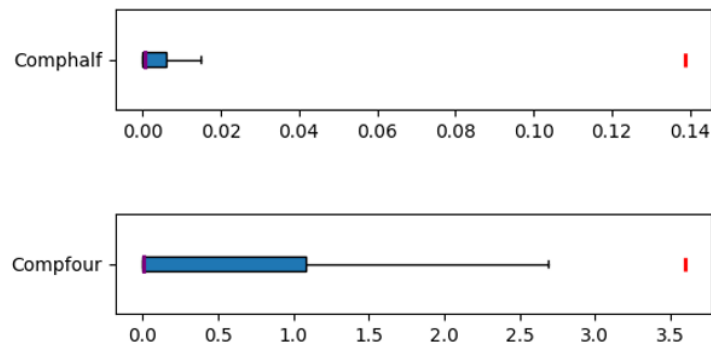


Figura 7.9: Boxplot Models Wallace Tree Compensador

Boxplot on es mostra la distribució d'error dels models Comphalf i Compfour. Línia púrpura correspon a la mediana i la línia vermella a la mitja. Elaboració pròpia.

7.1.3 Approximate Adders Wallace Tree

En el projecte s'han dissenyat dos tipus de models amb Approximate Adders que aproximen la generació de productes parcials. Es distingeixen segon la quantitat de Full Adders que s'han substituït en els sumadors de Wallace Tree de 8 i 16 bits. En concret, si s'ha utilitzat un 25% del AAn on "n" és un número del 1 al 12, el model s'anomena Wallace025AA_n i en el cas de que s'utilitzi un 50% del AAn, el model s'anomena Wallace050AA_n. En la següent seccions s'estudia quins són els Full Adders aproximats i la proporció que millor resultats han donat en les simulacions.

Wallace025AA_n

Els models que presenten menys consum són Wallace025AA₄ i Wallace025AA₈, tots dos aconseguen un estalvi per operació del 19,26%. Els models que han produït menys error han estat Wallace025AA₅, Wallace025AA₇ i Wallace025AA₁₂. Aquests tres circuits han operat sense error en les 150.000 multiplicacions de les simulacions. En canvi, tampoc han obtingut cap millora de rendiment.

Model	Ret(ns)	Con(mW)	Md.E.(%)	E.mx.(%)	Sp.UP(%)	Est.OP(%)
Base	59,043	207,47	-	-	-	-
Wallace025AA1	54,846	199,96	0,01759	0,08766	7,65	10,47
Wallace025AA2	51,673	200,61	0,11562	0,29352	14,26	15,38
Wallace025AA3	56,725	204,93	0,13881	0,32689	4,09	5,10
Wallace025AA4	52,993	186,64	0,01758	0,08622	11,42	19,26
Wallace025AA5	59,043	207,47	0,0	0,0	0,0	0,0
Wallace025AA6	51,385	208,43	0,23421	74,95595	14,90	12,57
Wallace025AA7	59,043	207,47	0,0	0,0	0,0	0,0
Wallace025AA8	52,993	186,64	0,01758	0,08622	11,42	19,26
Wallace025AA9	51,385	208,42	0,23421	74,95595	14,90	12,57
Wallace025AA10	51,385	208,43	0,23421	74,95595	14,90	12,57
Wallace025AA11	51,673	200,61	0,11562	0,29352	14,26	15,38
Wallace025AA12	59,043	207,47	0,0	0,0	0,0	0,0

Taula 7.10: Mètriques models AA 25% Wallace Tree.

Mètriques de rendiment i error dels models Approximate Adders en porporció de 25% que aproximen la suma de productes parcials del mòdul Wallace Tree de 24 bits.

Els dissenys que mantenen un equilibri entre optimització de rendiment i error introduït són Wallace025AA1, Wallace025AA4, Wallace025AA8 i Wallace025AA11.

Wallace050AA_n

Els models que presenten menys consum són: Wallace050AA8, Wallace050AA2 i Wallace050AA11. Encara que aquests dos últims presenten quartils petits però errors màxims excessius (al voltant del 75%). El model que sí que tenen un bon equilibri entre error i optimitzacions són els Wallace050AA4 i el Wallace050AA5.

Wallace Tree AA 25%

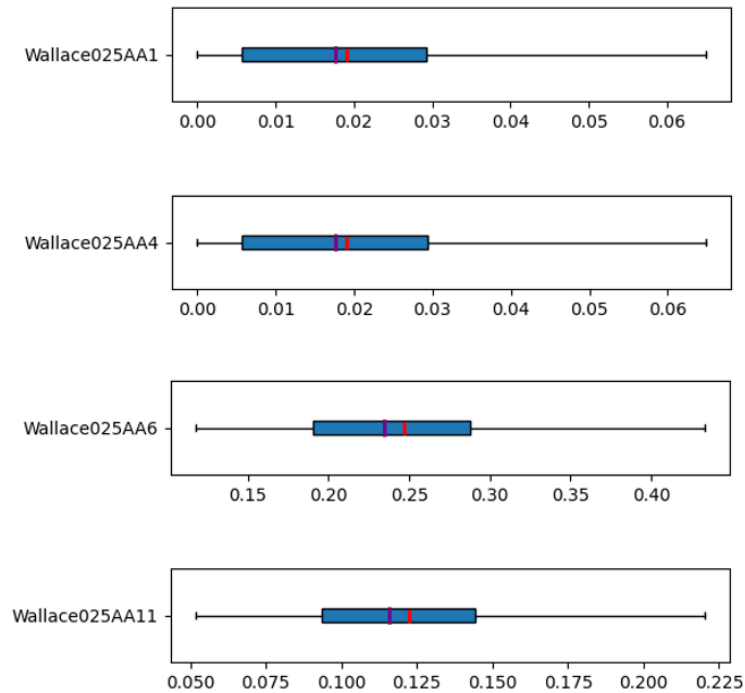


Figura 7.10: Boxplot Models Wallace Tree AA 25%

Boxplot on es mostra la distribució d'error dels models Wallace025AA1, Wallace025AA4, Wallace025AA6 i Wallace025AA11. Línia púrpura correspon a la mediana i la línia vermella a la mitja. Elaboració pròpia.

Model	Ret(ns)	Con(mW)	Md.E.(%)	E.mx.(%)	Sp.UP(%)	Est.OP(%)
Base	59,043	207,47	-	-	-	-
Wallace050AA1	55,971	187,46	0,26658	2,01357	5,49	14,35
Wallace050AA2	28,593	180,99	1,14195	74,80160	106,49	57,75
Wallace050AA3	54,937	184,88	1,94638	74,70011	7,47	17,09
Wallace050AA4	33,836	178,14	0,39820	2,17885	74,50	50,79
Wallace050AA5	48,816	194,42	0,22211	1,53291	20,95	22,52
Wallace050AA6	43,320	190,72	3,58457	74,53488	36,30	32,55
Wallace050AA7	56,619	188,34	0,39411	1,88096	4,28	12,95
Wallace050AA8	33,450	174,56	0,64746	2,20647	76,51	52,33
Wallace050AA9	54,945	196,84	3,51451	74,51294	7,46	11,71
Wallace050AA10	50,431	192,14	3,30732	74,52754	17,08	20,90
Wallace050AA11	28,593	180,79	1,47816	74,80160	106,49	57,80
Wallace050AA12	50,413	204,45	0,10355	1,30950	17,12	15,86

Taula 7.11: Mètriques models AA 50% Wallace Tree.

Mètriques de rendiment i error dels models Approximate Adders en porporció de 50% que aproximen la suma de productes parcials del mòdul Wallace Tree de 24 bits.

Wallace Tree AA 50%

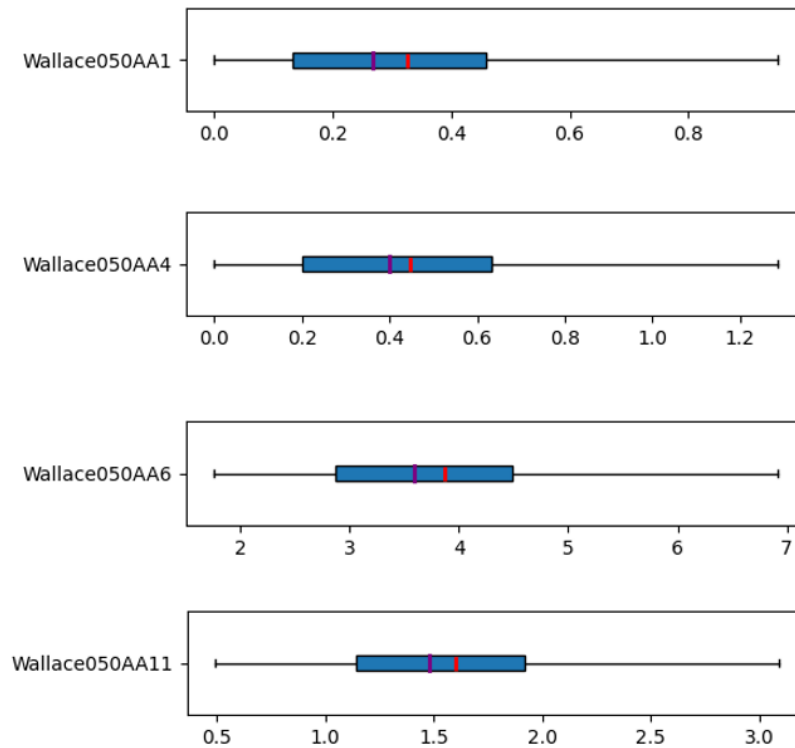


Figura 7.11: Boxplot Models Wallace Tree AA 50%

Boxplot on es mostra la distribució d'error dels models Wallace050AA1, Wallace050AA4, Wallace050AA6 i Wallace050AA11. Línia púrpura correspon a la mediana i la línia vermella a la mitja. Elaboració pròpia.

7.1.4 Estudi entre tipus

En les seccions anteriors hem comentat tots els punts forts i febles dels models aproximats dissenyats durant el projecte. Quan es compara dos models es pot utilitzar les mètriques per quantificar l'error i les millores. Però també és interessant fer una abstracció dels efectes generals que tenen certes tècniques d'aproximació en el càlcul de multiplicacions en coma flotant. En concret podem observar que:

- **Low OR:** Els models que utilitzen el sumador Low OR necessiten que el grau de nombre de portes OR superi el número de Full Adders perquè els dissenys presentin errors en el càlcul i optimització de rendiment. Aquesta tècnica millora els retards dels models Especulatiu i Compensadors però no els del model Segmentat. Aquest model obtenen de les millors relacions error produït/ optimització d'energia i temps. En aquest cas es pot observar que els valors obtinguts en les simulacions només varien en una quantitat baixa. Tot i això, l'estalvi d'energia es manté sobre el 20% i l'error amb una cota màxima inferior a l'1%.

- **Segmentat:** els models segmentats aconseguixen reduir el retard amb una magnitud igual i superior amb la qual ho fan els models Low OR, el speed up del model Segmentat 8 es dispara fins al 57,56% i es proclama com el model més ràpid que no utilitza Approximate Adders. En canvi, Segmentació 4, la seva versió amb menys segmentació, sí que competeix amb ADD OR 32 en termes d'optimització. Els dissenys segmentats tenen errors de magnitud molt superior. Si aquesta segmentació es dona en la generació de productes parcials, sí que s'obté un error molt més elevat i que es reparteix regularment amb una mitja del 0,52% i un error màxim que es dispara fins els 6,05%.
- **Especulatiu:** els models especulatius tenen un rendiment òptim en l'aproximació de la suma dels productes parcials. La reducció del retard és més discret que l'obtingut en els models Segmentats i Low OR. Aquest model, però, ha aconseguit reduir l'error dels models segmentats. Tot i això, l'error màxim només s'ha reduït al 6,03%. Si es parla de l'especulació aplicada a la generació de productes parcials, l'error és més important, pel model SpecFour, ha reduït el quartil 0,50 de 8,12% que hi ha obtingut SegFour a només 0,07%. En contra, la reducció no és tan significativa al quartil 0,75 i finalment SpecFour aconsegueix gairebé el triple d'error màxim (73,26%) en les simulacions.
- **Compensador:** els models compensadors redueixen per norma general l'error dels models Segmentats i Especulatius, especialment en la suma de productes parcials. En el cas de la generació de productes parcials, aquests models amb compensador d'error també redueixen les proporcions d'errors excepte en l'error màxim. Tot i això, les optimitzacions de rendiment són més important en els models Compensador 4 i Compensador 8.
- **Approximate Adders:** els models que utilitzen Full Adders aproximats són els dissenys que més error han generat. En el projecte, els circuits que aproximen la suma de productes parcials tenen errors d'una magnitud que no és comparable amb la resta de models. La raó és que es tracta dels únics models que han substituït tots els Full Adders per versions aproximades. Per tant, tota la mantissa generada és aproximada. En futurs objectius del projecte, seria interessant investigar si reduint la proporció d'aquesta aproximació es pot controlar la proporció d'error en els resultats. Especialment en models que han donat bons nivells d'estalvi d'energia i speed up, perquè són els que tenen més marge per ajustar, com per exemple AA4 i AA5. En canvi, sí que s'han estudiat diverses proporcions d'aproximació en la generació de productes parcials. Si s'han substituït el 25% dels Full Adders en els sumadors de Wallace Tree de 8 i 16 bits, els errors són de magnitud similar a la resta de models, com també són les seves optimitzacions. Gràcies a això, podem considerar circuits Wallace025AA2, Wallace025AA4, Wallace025AA8 i Wallace025AA11, rivals a batre per les altres tècniques d'aproximació. En el moment que aquesta proporció s'apropa al 50%, molts models amb bones mètriques d'optimització veuen eclipsat el seu funcionament per errors màxims que s'apropen al 75% d'error relatiu. Aquesta és el cas de Wallace050AA2 i Wallace050AA11 entre altres. Altres també són els casos de Wallace050AA8 i Wallace050AA12 que han aconseguit incrementar de forma substancial les optimitzacions amb un increment controlat de l'error.

Aquestes són conclusions que s'han extret mitjançant inducció sobre els resultats de les simulacions. Per a pròxims objectius del projecte, seria interessant estudiar si amb certes combinacions es pot aprofitar el grau d'optimització que han presentat els diversos models, o en un cas contrari, estudiar si combinar aquestes tècniques no suposen millora respecte a utilitzar només una.

7.2 Aproximacions de l'exponent

Les aproximacions en el càlcul de l'exponent són les que més risc comporten. El pes que tenen els bits de l'exponent és molt més gran que el pes que tenen els bits de la mantissa. Per tant els errors en l'exponent suposaran una desviació major en el resultat. Un altre factor que és perillós

en l'aproximació dels exponents és la conversió de tipus. En el capítol 1.1.2, es veu com l'estàndard IEEE-754 utilitza el valor de l'exponent per denotar nombres especials com l'infinít, NaN, etc.

En aquest projecte s'han dissenyat 3 models que aproximen el càlcul de l'exponent. Cadascun dels models aplica una de les següents tècniques: Segmentació, Especulació i Especulació amb compensació d'error. Els circuits reben el nom d'expoSeg, expoSpec i expoComp respectivament. A continuació els resultats que han obtingut en les simulacions:

Model	Ret(ns)	Con(mW)	Md.E.(%)	E.mx.(%)	Sp.UP(%)	Est.OP(%)
Base	59,043	207,47	-	-	-	-
expoSeg	56,069	205,54	0,0	1,76684E+74	5,30	5,92
expoSpec	53,256	200,12	0,0	1,76684E+74	10,87	13,00
expoComp	57,691	207,09	0,0	7,23701E+77	2,34	2,47

Taula 7.12: Mètriques models Exponent.

Mètriques de rendiment i error dels models que aproximen el càlcul de l'exponent.

Els models tenen una distribució de l'error molt interessant que demostra el comportament que és esperable d'aproximar l'exponent. Les medianes de tots els models es col·loquen al 0,0% d'error. Per una altra banda, els errors màxims es disparen fins de 1E74% i 7E77%. Això demostra que els errors en l'exponent són fatals per la precisió. El model expoSpec ha aconseguit reduir l'error d'expoSeg, però el sistema de compensació d'errors no ha estat efectiu en aquest cas i ha obtingut errors majors.

7.3 Aproximacions mòdul Normalitzador

El model Normal OR ha obtingut els resultats esperables segons a la modificació que s'ha realitzat. L'estalvi d'energia i speed up s'han vist incrementats però de forma discreta. Per norma general, els models que aproximen el mòdul Wallace Tree han obtingut millor mètriques d'optimització.

Model	Ret(ns)	Con(mW)	Md.E.(%)	E.mx.(%)	Sp.UP(%)	Est.OP(%)
Base	59,043	207,47	-	-	-	-
Norm OR	57,557	204,93	0,0	50,0	102,58	3,71

Taula 7.13: Mètriques model Normalització.

Mètriques de rendiment i error del model que aproxima el procés de normalització.

Els errors es presenten en proporcions molt disperses. Tenint en compte que només s'aproxima el bit de menys pes del exponent, la majoria d'operacions s'executen sense errors. En canvi, si en la suma de 1 es produeix carry cap el segon bit de menys pes, el error es dispara fins el 50%. Això fa que la mediana d'error sigui del 0,0% però el quartil 0,75 sigui del 50% d'error.

7.4 Conclusions de l'anàlisi

Durant aquest capítol s'han vist i estudiat els models aproximats. La gran quantitat de dissenys ha permès que s'obtinguin mètriques d'error d'una varietat molt gran de magnituds. Les optimitzacions de rendiment i consum també dispers entre models i conjuntament a l'error, fa que no sigui una decisió fàcil escollir quines són les millors.

Un fet destacable és que alguns models aconseguixen un error molt petit (per sota del 0,01% en el cas d'ADD OR 32) i unes mètriques de millora considerables. Aquest resultat seria interessant per aplicar a acceleradors els quals redueixin el temps de càlcul amb un error imperceptible. Són molts els models que acoten el seu error per sota de l'1%, encara que les modificacions en el circuit són considerables. Això és donat a la mateixa estructura del multiplicador. El mòdul Wallace Tree genera 48 bits de mantissa dels quals només 23 es reflectiran en el resultat. Per tant, la gran part dels bits de la part baixa no tenen efecte en el resultat. La possibilitat que un bit "carry" es propagui per diverses posicions és petita. Per tant, es pot concloure que les modificacions en la part baixa del Wallace Tree té un efecte molt petit en l'error del model. Aquest cas només es pot aplicar en el cas dels números normals, perquè la normalització només necessita els bits de la meitat més alta.

En la secció 7.1.4, s'estudia els resultats entre les diferents tècniques d'aproximació. Els sumadors Low OR necessiten un cert grau d'aproximació per començar a presentar errors, justament pel fet explicat anteriorment. Aquesta és la tècnica que per norma general menys error té.

Els models Segmentats, Especulatius i Compensats han obtingut resultats esperables. El mòdul especulador ha reduït l'error de les versions que no l'utilitzen. També ho ha fet el mòdul Compensador, la implementació ha funcionat correctament. El disseny dels circuits Especulador i Compensador no són fixes i la seva lògica també dona peu a modificacions. Sobre aquests tres models, els que obtenen menys error, també obtenen pitjor mètriques d'optimització. En el moment que aquestes aproximacions s'apliquen a la generació els errors són més importants i en alguns casos no s'obtenen gairebé millores.

Els Approximate Adders no han generat resultats bons resultats en la suma dels productes parcials. El problema aquí ha estat la proporció d'AA utilitzats que ha estat del 100%. Seria interessant reduir aquesta proporció per ajustar l'error. Els models que aproximen la generació de productes parcials obtenen millors errors i optimitzacions en les modificacions dels mòduls Wallace Tree de 8 i 16 bits. Aquest és el cas del model Wallace050AA8, que acota l'error per sota del 2,2% i té un speed up del 76,51% i un estalvi energètic per operació del 52,33%.

Un cas diferent han estat els models que no aproximen el Wallace Tree. Les aproximacions realitzades en els mòduls Exponent i Normalitzador no han produït models interessants. Han generat errors molt grans en un percentatge petit d'operacions i les seves mètriques d'optimització no són suficient bones per justificar aquest error. La inclusió d'aquests models en el projecte ha tingut com a fi fer un estudi més ampli.

En la figura 7.12, s'ha posicionat els models aproximats segons les mètriques de speed up i estalvi de consum energètic. Aquesta figura ens ajuda a observar la corba logarítmica que dibuixen els models.

Finalment, l'etapa d'anàlisi ha estat molt productiva. L'estudi dels resultats ha ajudat a comprendre quins han estat els efectes de les aproximacions en el càlcul de productes de coma flotant. Gràcies a les reflexions prèvies, aquest projecte o futurs projectes poden generar una segona ronda de simulacions on els models intentin millorar els detalls de models anteriors.

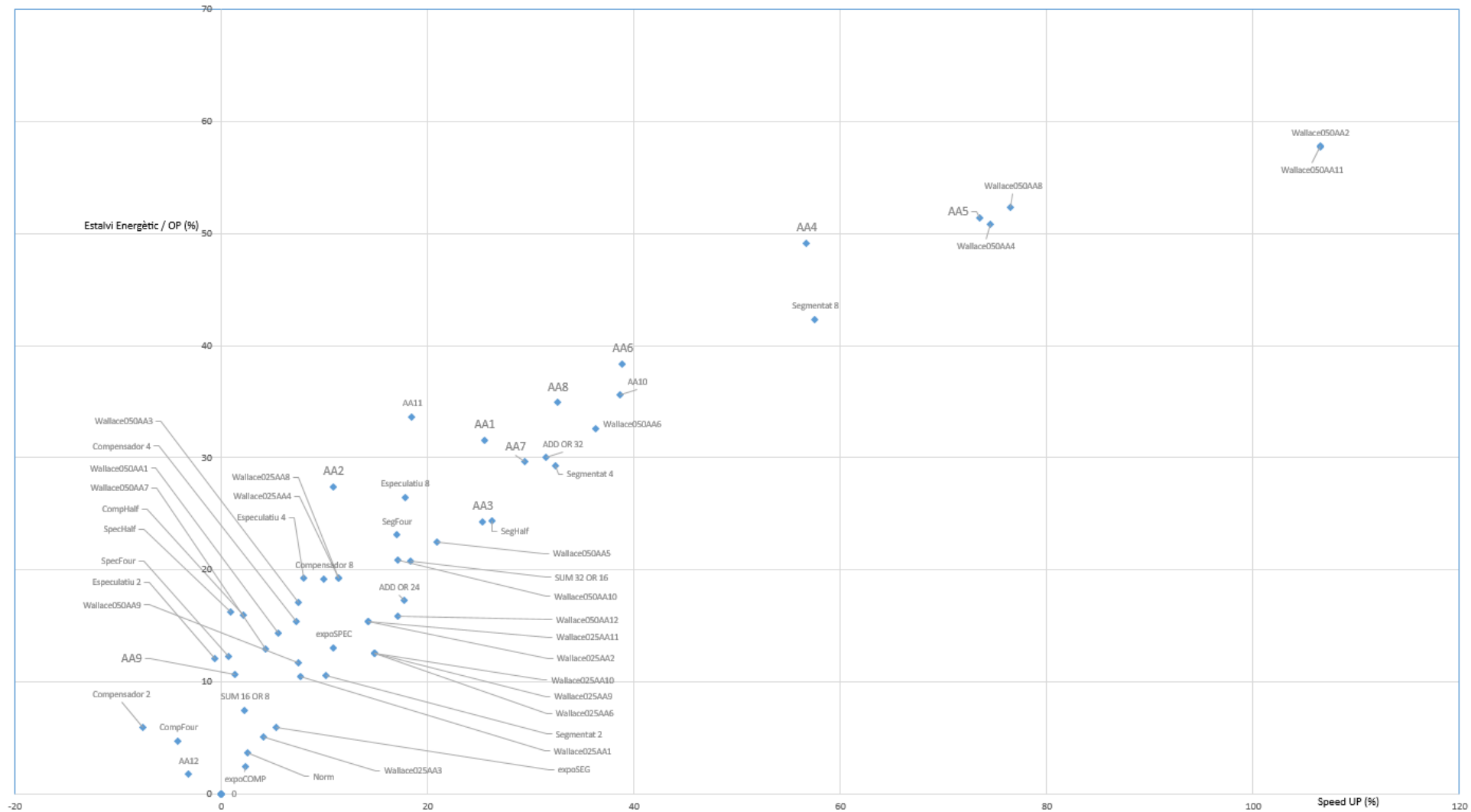


Figura 7.12: Optimitzacions de models aproximats.
 Gràfica on es localitza els models aproximats segons els seus nivells d'estalvi d'energia (eix y) i el speed up (eix x). Elaboració pròpia.

Capítol 8

Conclusions

El treball ha estat un camí ple de reptes. És un projecte interessant donat que combina en un bon equilibri l'estudi teòric de tècniques d'aproximació amb la seva aplicació en un circuit. Es va decidir estudiar quins són els aspectes d'aquestes modificacions en el producte de nombres de coma flotant. El format d'aquests nombres estan tipificats per l'estàndard IEEE-754. Aquest document ha indicat el format dels bits que seguia cada operand i un seguit de característiques que han ajudat la implementació. Seguir aquest estàndard també ha fet que el treball realitzat sigui aplicable a processadors reals o acceleradors.

El primer repte va ser dissenyar el primer model que calcula sense aproximacions. Les eines utilitzades requereixen una corba d'aprenentatge que era intimidant. Per sort, els coneixements previs obtinguts en altres assignatures han ajudat a fer aquesta tasca menys costosa. També ha estat satisfactori l'entorn de treball que s'ha creat especialment per aquest projecte. El sistema d'automatització de tasques ha facilitat que es pugui realitzar les simulacions dels 63 circuits aproximats. Dissenyar el sistema de verificació i tests especialment per aquest projecte també ens ha assegurat que es compleixen els requisits que experiments com aquests tenen.

L'estudi de les tècniques d'aproximació ha estat molt enriquidora. Són extretes de documents científics que busquen que comparteixen el mateix objectiu. Els resultats obtinguts en els articles s'han adaptat i aplicat al cas que es presentava. Integrar aquestes modificacions exitosament i oferir la varietat de solucions alternatives ha estat un dels resultats més interessants del projecte.

L'etapa de simulacions ha estat una de les etapes més interessants del projecte. Un cop els models estudiats s'havien compilat i verificat, es poden extreure les mètriques d'error, consum i retard. Aquestes dades són el propòsit de tot el treball realitzat. Amb aquestes dades s'ha pogut analitzar el comportament dels models aproximats i determinat la productivitat de les modificacions.

A l'inici del projecte es van establir uns objectius que s'han complert més que satisfactòriament. Tot i això, s'ha requerit ajustar la distribució del temps de treball per complir la data límit. Els possibles futurs objectius que l'autor considera són:

- **Estudiar aquestes tècniques en altres arquitectures:** l'estàndard IEEE-754 també especifica arquitectures de 16, 64 i 128 bits. Repetir l'estudi en aquests formats seria més directe que en un primer cas.
- **Combinar tècniques d'aproximació:** estudiar com les tècniques d'aproximació treballen quan es combinen seria un tema interessant.

- **Mòduls diferents:** el mòdul Wallace Tree ha estat el protagonista de 59 de les 63 aproximacions. Existeixen alternatives al disseny Wallace Tree que també es podrien estudiar com aproximar-se. Un exemple és el Dadda Tree [5].
- **Profunditzar més en les característiques del producte de coma flotant:** aquest projecte ha estat un primer contacte amb el disseny d'una unitat funcional aproximada. Per futurs projectes, seria interessant aplicar el procés en un disseny real i un cas similar on treballen els arquitectes de computadors professionals.
- **Altres operacions:** l'entorn de treball i tècniques d'aproximació que s'han utilitzat per al producte de coma flotant es podria utilitzar en altres operacions i estudiar el seu comportament tal com s'ha fet en aquest projecte.

Finalment es pot concloure que les unitats funcionals aproximades es presenten com una opció emergent per dissenyar processadors més eficients en consum i temps. És possible que en els pròxims anys veiem solucions similars aplicar-se en àmbits com: algorismes de processament d'imatges, sistemes a temps real, Internet of Things, sistemes encastats amb sensors... Serà responsabilitat dels Arquitectes de Computadors superar el repte que suposa una societat que cada dia més informatitzada i on els recursos naturals s'esgoten. Repte que se superarà amb noves tecnologies i enginy.

Bibliografia

- [1] Agent. <https://www.infojobs.net/barcelona/tecnico-hardware-senior/of-id3003764cd4f689f68020db327488f?applicationOrigin=search-new&page=1&sortBy=RELEVANCE>. Accessed: 2020-10-12.
- [2] Aigua. <https://www.ticbeat.com/empresa-b2b/cuanto-pagas-por-el-agua-segun-la-comunidad-autonoma-en-la-que-vivas/>. Accessed: 2020-10-12.
- [3] Branch predictors. https://en.wikipedia.org/wiki/Branch_predictor.
- [4] Comparacions. <http://puntoflotante.org/errors/comparison/>.
- [5] Dadda multiplier. https://en.wikipedia.org/wiki/Dadda_multiplier.
- [6] Drac. <https://drac.bsc.es/>. Accessed: 2020-09-28.
- [7] Drac socis. <https://drac.bsc.es/ca/consorci/socis>. Accessed: 2020-10-12.
- [8] An efficient implementation of floating point multiplier. https://www.researchgate.net/publication/224242069_An_efficient_implementation_of_floating_point_multiplier. Accessed: 2020-12-04.
- [9] Electricitat. <https://www.elperiodico.com/es/economia/20151113/cuanto-paga-al-mes-en-el-recibo-de-la-luz-4669377>. Accessed: 2020-10-12.
- [10] Error en arrodoniment. <http://puntoflotante.org/errors/rounding/>. Accessed: 2020-12-04.
- [11] Exponent. <https://es.wikipedia.org>. Accessed: 2020-12-03.
- [12] Ieee. <https://www.ieee.org/about/ieee-history.html>. Accessed: 2020-09-28.
- [13] Ieee. https://es.wikipedia.org/wiki/IEEE_754. Accessed: 2020-09-28.
- [14] Llei d'amdahl. https://en.wikipedia.org/wiki/Amdahl%27s_law.
- [15] Lley de moore. https://es.wikipedia.org/wiki/Ley_de_Moore. Accessed: 2020-09-28.
- [16] Moore graphic. <https://github.com/karlrupp/microprocessor-trend-data/blob/master/48yrs/48-years-processor-trend.png>. Accessed: 2020-09-28.
- [17] Mètodes d'arrodoniment. https://en.wikipedia.org/wiki/IEEE_754-1985#Rounding_floating-point_numbers.
- [18] Powerplay power analyzer. <https://www.intel.com/content/www/us/en/programmable/documentation/xhv1529966780595.html>.
- [19] Unitat de coma flotant - floating-point unit. https://es.qwe.wiki/wiki/Floating-point_unit. Accessed: 2020-09-28.

- [20] Vhdl testbench. <https://allaboutfpga.com/vhdl-testbench-tutorial/>.
- [21] P. Deepa G. Anusha. Design of approximate adders and multipliers for error tolerant image processing. *www.elsevier.com*, (102940), 2019.
- [22] Y. Manoj Sai Teja D. Akhila K. Nikhil Sai P. Sravan Kumar K. Hari Kishore, B. K. V. Prasad. Design and comparative analysis of inexact speculative adder and multiplier. *International Journal of Engineering Technology*, (7):413–418, 2018.