# Resum

En aquest projecte s'introdueix el concepte del PUF (*physical unclonable function*), un element de seguretat amb la funció de protegir als circuits integrats contra possibles atacs maliciosos. Un PUF és una funció física incorporada en una estructura física, que és fàcil d'avaluar però difícil de predir. La següent analogia permet entendre, de forma planera, la funcionalitat d'aquest sistema: un PUF en un circuit integrat té la intenció de realitzar la mateixa funció que les empremtes dactilar en les persones. L'empremta digital permet la identificació d'una persona i un PUF té la funció d'identificador de circuit integrats. Un dels camps d'aplicació d'aquest dispositiu és el de les targetes de crèdit, que necessiten nous sistemes de seguretat per a evitar la seva falsificació.

Des de l'any 2004 fins a la actualitat s'ha dedicat un gran esforç en investigació i recerca sobre aquest sistema, i s'han presentat moltes arquitectures diferents de PUF. Un dels darrers PUFs que han sorgit és el BR-PUF (*bistable ring* PUF) [1], tot i que té algunes propietats molt bones, presenta dos grans inconvenients a solucionar. Per una banda, aquest sistema és oscil·lant, amb un temps d'estabilització llarg [2]. A més, en alguns casos, el sistema genera una resposta poc uniforme i fàcil de predir [3].

Per a analitzar aquestes problemàtiques, aquest projecte presenta un model matemàtic que permet estudiar el comportament dels BR-PUFs. El model matemàtic obté les dades estadístiques per a caracteritzar el sistema de forma més àgil que les simulacions basades en l'ànàlisi dels components elèctrics (SPICE). Mitjançant dues modificacions presentades al projecte, s'aconsegueixen solucionar els problemes d'aquests circuits electrònics. El filtre IOCF (*interleaved oscillation canceller filter*) [4] permet suprimir les oscil·lacions transitòries del sistema. En el projecte també s'introdueix la arquitectura TBR-PUF (*twisted* BR-PUF) [3] que és capaç d'uniformitzar la resposta davant a variacions locals del procés de fabricació.

S'ha dissenyat un prototip de BR-PUF amb filtre IOCF per a ser implementat en un ASIC (*application-specific integrated circuit*) per a comprovar el funcionament experimental del circuit electrònic. El disseny del prototip s'ha realitzat utilitzant la tecnologia CMOS de 65 nm de ST.

# Sumari



ETSEIB

# 1. Glossari

**PUF:** *Physical unclonable functio*. És una funció incorporada en un dispositiu físic. Aquesta funció té la propietat de ser fàcilment avaluable, però difícilment predictible.

***Challenge:*** Variable independent, o valor d'entrada del PUF.

***Response:*** Variable dependent, o valor se sortida del PUF.

**CRP:** *Challenge response pair*. Binomi format per un *challenge* i una *response.*

**BR-PUF:** *Bistable ring* PUF. Arquitectura de PUF basada en la utilització d'anells d'inversors biestables.

**IOCF:** *Interleaved oscillation canceller filter*. Filtre que suprimeix les oscil·lacions transitòries en anells d'inversors biestables.

**TBR-PUF:** *Twisted* BR-PUF. Modificació del BR-PUF que permet solucionar el problema de la uniformitat aleatòria.

**Cadence virtuoso:** Programari de disseny de circuit electrònics. Permet simular circuit electrònics mitjançant l'estàndard SPICE.

**Simulacions SPICE:** Simulacions analògiques de circuits electrònics. Permet comprovar el funcionament del circuit.

**Simulacions Monte Carlo:** Simulacions que consisteix en afegir variacions estadístiques aleatòries en les propietats del circuit a simular per tal de comprovar l'efecte que tenen les variacions de procés de fabricació sobre el circuit.

**ASIC:** *Application-specific integrated circuit*. Circuit integrat dissenyat per a realitzar una funció concreta.

**FPGA:** *Field progrommable gate array*. Dispositiu semiconductor amb lògica programable.

**Pad:** Àrea d'un circuit integrat destinada a comunicar el xip amb l'exterior.

# 2. Prefaci

L'ús i implementació de la criptografia en els circuit integrats és un àmbit de recerca en progrés continu, ja que resulta un tema crucial per a sistemes on la privacitat de les dades és crítica, com per exemple en les targetes de crèdit. Una de les raons principals de la investigació en aquest tema és que, a mesura que sorgeixen noves tecnologies que milloren la seguretat, també es desenvolupen noves estratègies que permeten superar aquestes. Això va acompanyat del perfeccionament de les tècniques d'enginyeria inversa que han proliferat en els darrers anys degut al valor de la informació emmagatzemada en aquests dispositius.

L'atac físic contra dispositius criptogràfics usualment aprofita fuites d'informació del propi dispositiu que es pot recuperar mitjançant tècniques de *side channel attack* [5]. Aquestes llegeixen i analitzen informació, com per exemple el consum energètic o la radiació electromagnètica per tal d'obtenir informació privada. Una altra forma d'extreure la informació dels dispositius és provocant errors en la computació d'algoritmes, per mitjà de tècniques anomenades *fault injection attacks* [6] que trenquen el paradigma de capça negra dels sistemes criptogràfics i en conseqüència corrompeix les garanties de seguretat dels algorismes.

Prevenir o detectar aquest atacs s'ha convertit en un dels punts claus dins de la criptografia. En aquest context apareix el concepte de *Physical Unclonable Function* (PUF), que té la intenció de millorar la seguretat en dispositius criptogràfics, complementant les construccions criptogràfiques clàssiques. En concret un PUF té la funció bàsica d'emmagatzemar una clau secreta a l'interior d'un dispositiu (que usualment sol ser electrònic).

En els darrers anys han sorgit moltes arquitectures aptes per a ser usades com a PUF. Una de les més recents és el BR-PUF que basa el seu funcionament en la utilització d'anells d'inversors biestables.

## 2.1. Origen del projecte

L'origen d'aquest projecte es troba en la col·laboració que vaig realitzar al departament d'enginyeria electrònica de l'ETSEIB l'any 2013. Durant la realització d'aquesta tasca vaig tenir la possibilitat de col·laborar amb l'institut Fraunhofer de Munich (AISEC) en la simulació, a nivell de transistors, de PUFs. A partir d'aquest treball, se'm va presentar l'oportunitat de realitzar una estada a l'institut AISEC durant l'estiu de l'any 2013. En aquesta estada vaig treballar en la investigació sobre els BR-PUFs, creant models matemàtics per al seu anàlisi i estudiar la problemàtica existent en aquests.

# 3. Introducció

## 3.1. Objectius del projecte

L'objectiu principal d'aquest projecte és la realització d'un prototip amb un disseny experimental del BR-PUF per a la seva experimentació. Però per a l'acompliment d'aquest objectiu cal realitzar un estudi previ dels PUFs. D'aquesta manera els objectius del projecte són:

- **Estat de l'arts del PUFs.** Introducció del concepte i funcionament d'un PUF genèric. Presentació de les diferents arquitectures des PUFs que han sorgit en els darrers anys.
- **Estudi del BR-PUF.** Estudi del circuit utilitzant programes de simulació SPICE per a l'estudi el comportament d'aquest circuit. Creació d'un model matemàtic del BR-PUF per a la seva caracterització. Presentació de millores pel comportament del circuit electrònic.
- **Disseny d'un prototip de BR-PUF amb el filtre IOCF** mitjançant la tecnologia CMOS de 65 nm de ST per a la realització d'experiments sobre el circuit.

## 3.2. Abast del projecte

Aquest projecte es centra en el disseny d'un prototip del circuit electrònic d'un BR-PUF amb filtre IOCF. Prèviament al disseny del prototip, es realitza una exposició de l'estat del art dels PUFs, mostrant, tant les seves característiques, com algunes de les seves tipologies més conegudes.

Per al disseny d'aquest circuit, primerament s'estudia el sistema mitjançant la creació d'un model matemàtic. Utilitzant el llenguatge de programació Python, i les seves llibreries científiques, s'analitza el model creat. Es valida el correcte funcionament del circuit electrònic i del model matemàtic realitzant simulacions SPICE del circuit. Finalment, es realitza el disseny del prototip a nivell de *layout* per a ser implementat en un circuit integrat ASIC.

Per a controlar adequadament el prototip s'ha creat un autòmat de control en codi VHDL per a ser implementat en una FPGA. Aquest autòmat té la funció de gestionar les senyals d'entrada i sortida del prototip per a que aquest funcioni correctament.

# 4. PUF: Funció no clonable físicament

Una funció no clonable físicament (PUF) [7] és una funció integrada en un objecte físic, com per exemple un circuit integrat. Aquesta funció ha de ser fàcil d'avaluar, però difícil de predir. A l'aplicar un senyal d'entrada (anomenat *challenge*) al PUF, aquest genera una certa resposta (*response*). Cada grup de *challenge-response* rep el nom de parell *challenge-response* (CRP). Perquè aquesta funció sigui no clonable (com indica el seu nom), la *response* ha de dependre tant del *challenge* com de les propietats físiques úniques del dispositiu que conté el PUF. Aquestes propietats físiques han de dependre de les variacions del procés de fabricació del dispositiu que conté el PUF. Es a dir que, aprofitant el fet que no es poden construir dos productes idènticament iguals, el conjunt de *responses*, obtingudes a través del mateix conjunt de *challenges*, no serà igual en dos dispositius aparentment idèntics.

A partir d'aplicar una sèrie de *challenges* i analitzar el conjunt de *responses* obtingudes es pot generar una clau secreta que identifiqui el PUF. Aquesta clau no s'assigna de forma arbitrària al dispositiu, sinó que es genera de forma incontrolada durant la seva fabricació gràcies a les inevitables variacions del procés. Per tant, cada PUF tindrà una clau diferent. Els camps d'aplicació dels PUFs són, entre d'altres, l'emmagatzematge implícit de claus secretes, la identificació de dispositius o la generació de claus criptogràfiques.

## 4.1. Història dels PUFs

Al llarg de la història hi han molts exemples en els quals la criptografia i la possibilitat d'emmagatzemar una clau en un dispositiu han tingut gran rellevància. Per exemple, les forces militars d'Alemanya van utilitzar, a partir del 1930, una màquina criptogràfica anomenada Enigma. Aquest dispositiu tenia la funció d'encriptar missatges mitjançant mecanismes electromecànics. La desencriptació dels missatges era molt complicada sense una altra màquina Enigma. Els nazis utilitzaven l'Enigma per transmetre missatges confidencials durant la segona guerra mundial. Gracies a un gran esforç en criptoanàlisi, els aliats van poder desxifrar la codificació de l'Enigma i en conseqüència espiar els missatges enemics codificats. Aquest fet els va permetre tenir certs avantatges durant el transcurs de la guerra.

La idea d'utilitzar propietats físiques aleatòries per identificar objectes tampoc és nova. Per exemple, la biometria ha utilitzat, des del segle XIX, les empremtes digitals per a identificar persones. Però no es fins l'any 2001 en el qual Pappu introdueix i formalitza la base del que serà el PUF en la seva tesi "Physical One-Way Functions" [8] que proposa identificar certs dispositius per les variacions aleatòries en les seves propietats físiques, en comptes

d'assignar un codi d'identificació arbitrari. A aquest dispositiu el va anomenar *Physical One-Way Function* (POWF). El concepte inicial era utilitzar una fitxa transparent dopada de forma aleatòria amb partícules que dispersessin la llum. Per diferenciar una fitxa d'una altra, s'observava el patró que es creava al fer incidir un raig làser sobre aquestes.

Degut l'elevada sensibilitat del sistema d'orientació del raig làser, i a les dificultats derivades de les grans restriccions mecàniques del sistema proposat per Pappu, es van acabar inventant els anomentats PUFs. La base del PUF és la mateixa que el POWF, es a dir, utilitzar variacions de procés d'un dispositiu per identificar-lo, però en el cas dels PUFs s'aprofiten les variacions produïdes en el procés de fabricació d'un circuit integrat. Aquesta tendència es va iniciar l'any 2004 quan Lim publica la tesi "Extracting secret keys from integrated circuits" [9] on proposava un dispositiu anomenat *Arbiter-PUF*. Aquest sistema utilitzava les variacions en el temps de retard de les portes lògiques com a identificador d'un xip. Incorporant el PUF a l'interior d'un circuit integrat, es generen els *challenges* i les *responses* a l'interior del mateix dispositiu, disminuint els errors deguts a les pertorbacions externes. A partir d'aquesta publicació [9], la integració dels PUFs en dispositius electrònics s'ha expandit i s'han desenvolupat, de forma gaire bé anual, noves estructures per implementar PUFs.

Típicament els PUFs integrats en circuits electrònics es basen en aprofitar certes característiques dels circuits electrònics que varien de forma aleatòria i incontrolada en el procés constructiu d'aquest dispositius. Exemples d'aquestes propietats són la tensió llindar dels transistors, les capacitats de les vies metàl·liques del circuit, el retard de certes portes lògiques o l'estat al qual tendeix una cel·la de memòria volàtil després del *power-up*.

## 4.2. Aplicacions dels PUFs

Com ja s'ha comentat un PUF té dos aplicacions fonamentals, la primera i més estesa és la d'identificador de sistemes (generalment circuits integrats) i la segona és la generació de claus secretes i aleatòries. Seguidament es presentaran breument aquestes dues aplicacions.

### 4.2.1. PUF com a identificador de sistemes

Degut a que es dissenyen els PUFs de manera que tinguin una naturalesa no clonable i difícil de predir, aquest es pot utilitzar com a identificador de xips [10]. D'aquesta manera, un PUF en un circuit integrat es podria entendre com un dispositiu amb la mateixa funció que una empremta digital per a les persones. En altres paraules, aquesta aplicació dels PUFs consisteix en un mecanisme que permet prevenir la falsificació de xips.

Aquesta aplicació consta de les 2 fases mostrades en les figures 4.1 i 4.2. En una primera fase, anomenada enregistrament, s'obté un conjunt de CRPs i s'emmagatzemen en una base de dades CRP. En una segona etapa, anomenada verificació, s'aplica un *challenge* que estigui contingut en la base de dades CRP, i es compara la *response* obtinguda amb la continguda a la base de dades. Si la *response* obtinguda és suficientment pròxima a la emmagatzemada a la base de dades CRP es confirma la identificació positiva del dispositiu. Com que pot haver-hi certa variabilitat en la *response* d'un PUF cal definir una diferència llindar entre les *responses* per tal de minimitzar els errors deguts a identificacions i rebutjos falsos.

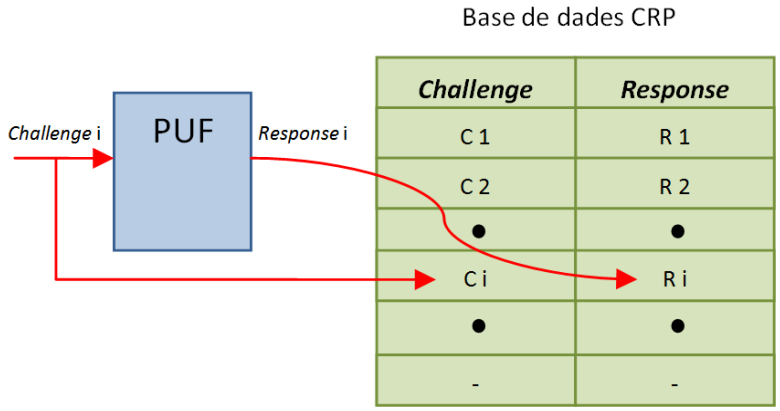


Figura 4.1: Fase d'enregistrament d'un PUF. S'emmagatzemen el *challenge* i la *response* generada pel PUF en una base de dades CRP

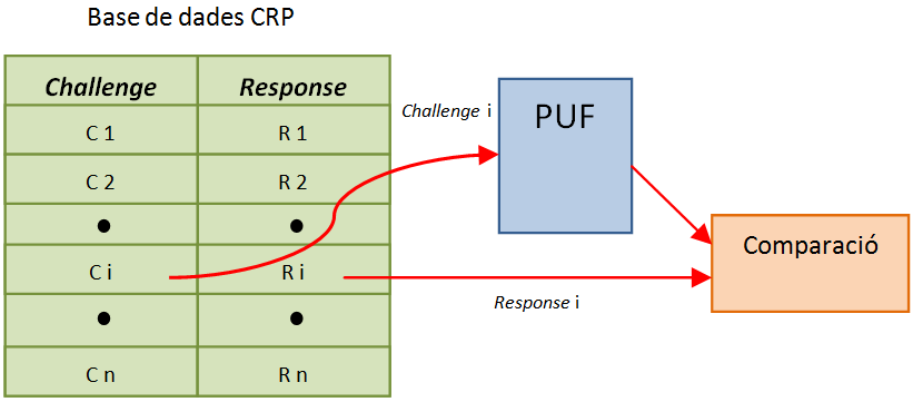


Figura 4.2: Fase de verificació d'un PUF. Es torna a aplicar el *challenge* al PUF per comprovar que la *response* obtinguda és idèntica a la de la base de dades CRP

### 4.2.2. Generació de claus secretes

Enviant una sèrie de *challenges* a un PUF s'obtenen un conjunt de *responses* que es poden seleccionar com a claus, en aquest cas fem servir el PUF com a generador de claus [11]. Degut a que usualment la *response* d'un PUF té associada soroll que la pertorba, és necessari fer un post procés d'estabilització de la clau. Usualment aquesta etapa de processament consisteix en un algoritme que combina les CRPs del PUF amb informació auxiliar emmagatzemada externament i que en conseqüència augmenta la fiabilitat del seu funcionament.

La utilització de PUFs, en circuits integrats, com a generadors de claus presenta propietats interessants, ja que en aquests sistemes la clau depèn de la variabilitat introduïda en el procés constructiu. Es a dir no hi ha cap etapa del disseny que de forma explicita introdueixi la clau en el dispositiu. A més, pel fet d'emmagatzemar la informació a les propietats físiques del xip, no hi ha cap element de memòria no volàtil que emmagatzemi de forma explícita la clau. D'aquesta manera s'obté una protecció addicional contra atacs físics al dispositiu electrònic, especialment els que empren enginyeria inversa.

## 4.3. Tipologies de PUFs

Les diferents propostes de PUF es poden classificar en funció del seu principi constructiu i funcional.

### 4.3.1. PUFs no elèctrics

Abans d'aparèixer els PUFs en els circuits integrats, es van crear una sèrie de sistemes de naturalesa no electrònica que tenien la funció d'identificar certs dispositius. Aquests s'aprofitaven de les variacions en propietats físiques no elèctriques per donar variabilitat a la resposta del PUF. A continuació es descriuen dos exemples d'aquests sistemes.

#### 4.3.1.1. *Optical* PUF

Un sistema basat en la utilització d'una fitxa transparent dopada amb microesferes refractives és considerat com l'inici de la recerca sobre els PUFs [8]. Es fa incidir un raig làser sobre la fitxa amb una certa orientació (que és el *challenge*). Com que la distribució de les microesferes en la fitxa és aleatòria, el patró que s'observarà en una càmera (que és la *response* del PUF) variarà d'una fitxa a l' altra. A més el patró capturat per la càmera també dependrà de la orientació d'aquesta.

#### 4.3.1.2. RF-DNA

De forma similar al *Optical* PUF, aquesta estructura [12] parteix d'una fitxa de material aïllant a la qual s'hi introdueixen petits cables de coure de forma aleatòria. Mesurant la dispersió de les ones electromagnètiques mitjançant un conjunt d'antenes es pot observar un patró únic per a cada fitxa que permet identificar-les.

### 4.3.2. PUFs analògics

Aquesta categoria engloba els PUFs que tenen com a operativa la mesura analògica d'una propietat elèctrica [13]. La propietat elèctrica mesurada tendeix a variar en el procés constructiu del circuit integrat. Actualment estan en desús a causa de la dificultat d'obtenir un elevat nombre de CRPs en comparació amb els PUFs electrònics

#### 4.3.2.1. $V_t$ PUF

Durant la producció d'un transistor, la seva tensió llindar ($V_t$) pot tenir petites variacions. Construint una estructura formada per varis transistors que alimenten a una resistència [10], es pot identificar un circuit electrònic mesurant la diferència de potencial produïda en a la resistència. Degut a que la tensió llindar pot tenir variacions en dos transistors dissenyats idènticament, la intensitat que circularà per la resistència de carrega no serà sempre igual, fent variar la tensió en la resistència.

#### 4.3.2.2. *Power distribution* PUF

Les variacions de procés poden produir variacions en els valors de les resistències d'un circuit integrat. Mesurant els valors d'un conjunt de resistències de un circuit electrònic es pot arribar a identificar un xip, fent que aquesta estructura es comporti com un PUF [14].

#### 4.3.2.3. *Coating* PUF

De forma anàloga a la estructura anterior, mesurant les capacitats paràsites entre pistes d'un circuit integrat, es pot arribar a identificar un xip [15]. Aquest fet es possible gràcies a que aquestes capacitats també varien de forma aleatòria en el procés constructiu.

### 4.3.3. PUFs electrònics basats en retards intrínsecs

En tot sistema electrònic real existeixen uns certs retards en les portes lògiques que el formen. A través de la creació d'un circuit que permeti mesurar aquesta propietat (que varia en el procés constructiu del circuit integrat), es pot identificar el sistema. Els PUFs dins d'aquesta categoria es basen en aquest principi per emmagatzemar una clau.

#### 4.3.3.1. *Arbiter* PUF

El principi de funcionament d'aquest PUF [16] és crear una carrera digital entre dues pistes del circuit integrat. S'emet un senyal per les pistes, i al final d'aquestes hi ha un circuit *arbiter* que decideix quin des dos senyals ha arribat més aviat. Aquest circuit *arbiter* genera una *response* en funció de la pista més ràpida. Els retards de les pistes es veuen afectats per les variacions de procés, fent que dues pistes iguals no tinguin exactament el mateix retard. Per afegir complexitat al sistema, es divideix la pista en trams afegint multiplexors i demultiplexors que creuen les pistes en funció del *challenge*. D'aquesta manera, per a un mateix parell de pistes es poden obtenir varies *responses*. A la figura 4.3 es mostra de forma esquemàtica el funcionament d'aquest tipus de PUF.

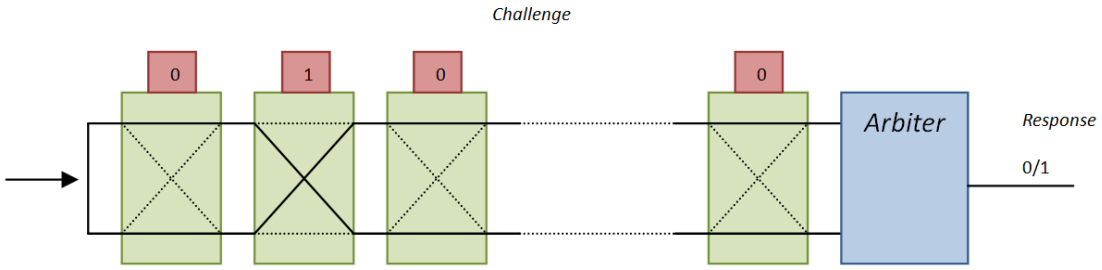


Figura 4.3: Arquitectura d'un *Arbiter* PUF. El *challenge* selecciona en quin tram es creuen les pistes. La *reponse* s'obté en funció de la pista més ràpida

Aquest sistema presenta l'avantatge de que amb un petit nombre d'etapes es pot obtenir un elevat nombre de CRPs. Per a $n$ etapes (es a dir $n$ *challenge*), aquest dispositiu pot generar fins a $2^n$ CRPs diferents.

No obstant això, aquesta estructura té el problema de que la seva *response* es pot predir fàcilment. Provant un conjunt de parells *challenge-response*, es pot crear un model matemàtic [17] que predigui la *response* d'un *chellenge* nou amb una elevada probabilitat. Aquest fet es degut a que els retards en cada un dels blocs del *Arbiter* PUF són additius, es a dir que el retard total d'una pista és igual a la suma del retard de cada un dels trams que la formen.

#### 4.3.3.2. *Ring Oscillator* PUF

Un anell oscil·lador és una estructura formada per un nombre senar d'inversors que estan disposats en forma d'anell (la sortida d'un inversor és la entrada del següent). Aquesta estructura té la propietat de que a l'alimentar-la tendeix a oscil·lar a una freqüència que depèn dels nombre de inversors i de les variacions de procés que afecten als inversors. Un *Ring Oscillator* PUF [18] és una estructura formada per un grup d'anells oscil·ladors

dissenyats idènticament, però que tenen freqüències diferents. Per obtenir la *response* d'aquests PUFs es compara la freqüència de dos anells oscil·ladors ,seleccionats a partir d'un *challenge* , indicant quin dels dos té una freqüència major. A la figura 4.4 es mostra l'esquema del *Ring Oscillator* PUF.

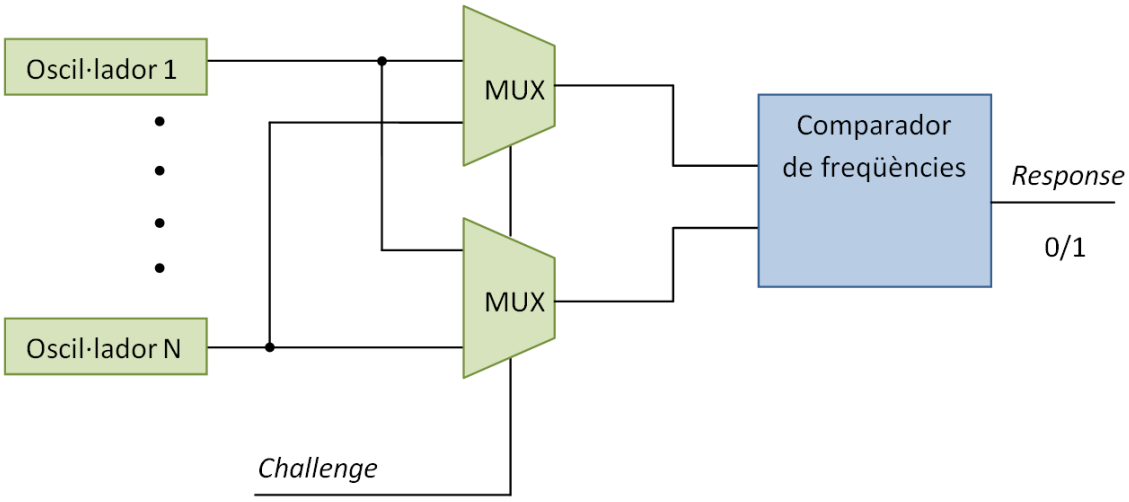


Figura 4.4: *Ring Oscillator* PUF. El *challenge* selecciona els oscil·ladors a comparar i es genera una *response* en funció del més ràpid

Un dels inconvenients d'aquest dispositiu és que per a obtenir un elevat nombre de CRPs cal un elevat nombre d'anells oscil·ladors [19]. Per tant cal una àrea de silici bastant gran per implementar el sistema.

### 4.3.4. PUFs electrònics basats en cel·les de memòria

Una cel·la de memòria d'un circuit digital té més d'un estat estable al qual pot arribar. Els PUFs d'aquesta categoria es basen en aprofitar la tendència de les cel·les de memòria per assolir un dels seus possibles estats estables.

#### 4.3.4.1. SRAM PUF

Una cel·la de memòria SRAM és una construcció formada per dos inversors realimentats, tal i com es mostra en la figura 4.5. Constructivament s'utilitzen 6 transistors, 4 per realitzar els inversors i 2 per les senyals de *read* i *write*.

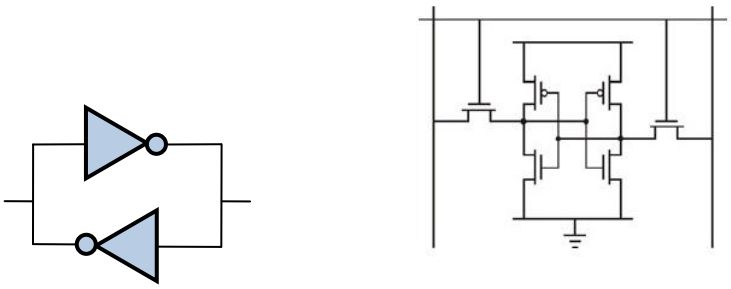

Figura 4.5: cel·la de memòria SRAM

Com que la sortida d'un inversor està connectada a l'entrada de l'altre, la cel·la de memòria SRAM té dos possibles estats estables ([1,0] o [0,1]). Si un inversor genera un '1' , l'altre inversor rep aquest valor com a entrada, generant un '0' a la seva sortida. Aquest '0' arriba al primer inversor estabilitzant el sistema, ja que el primer inversor segueix generant un '1'.

S'observa que, després d'alimentar la cel·la, aquesta tendeix cap a un estat o un altre de forma imprevisible. L'estat al qual arriba depèn de desajustos en els inversors produïts en el procés productiu de la cel·la . A més, s'observa que una cel·la tendeix cap al mateix estat pràcticament sempre, donant robustesa al sistema. D'aquesta manera, si es tenen el suficient nombre de cel·les SRAM, es pot crear un sistema que tingui el comportament d'un PUF [20]. En aquest cas el *challenge* és la selecció de la cel·la SRAM que s'està avaluant, i la *response* és l'estat al qual tendeix la cel·la. Per tant, si es vol un cert nombre de CRPs, es necessitarà el mateix nombre de cel·les SRAM.

#### 4.3.4.2. *Butterfly* PUF

Implementar un SRAM PUF en una FPGA és una tasca complexa ja que les FPGA inicialitzen les cel·les SRAM a 0. Per tal de solucionar aquest problema s'imita el comportament d'una cel·la SRAM, realimentant dos registres tipus *latch* [21]. El funcionament es basa en inicialitzar el sistema a un estat inestable i deixar que aquest evolucioni espontàniament cap a un dels dos possibles estats estables.

## 4.4. Propietats dels PUFs

### 4.4.1. Descripció de les propietats d'un PUF

Com ja s'ha comentat, una instancia d'un PUF és una funció integrada en un dispositiu físic, que a l'aplicar-li unes entrades, genera unes sortides. Es pot expressar la funcionalitat dels PUFs amb la següent notació matemàtica [4]:

$$\Pi : \chi \rightarrow \Upsilon : \ \Pi(C) = R \qquad \text{(eq 4.1)}$$

On $\Pi$ és una representació abstracta de la funció del PUF, que a partir de un *challenge* $x$, genera una *response* $y$. $\chi$ representa el conjunt de *challenge* possibles, i $\Upsilon$ el de *responses* possibles.

A partir d'aquesta notació es poden definir les següents propietats dels PUFs:

- Avaluable: donat un *challenge*, $C$, es pot avaluar la funció $\Pi(C)$ per obtenir una *response* $R = \Pi(C)$. Es a dir, que en un PUF sigui possible obtenir CRPs amb un temps i esforç reduït, tenint en compte les restriccions associades al dispositiu que el conté.

- Unicitat: $\Pi(C)$ conté informació sobre la identitat del dispositiu on esta incorporada la implementació física de $\Pi$. Mitjançant la obtenció d'un conjunt de CRPs, s'hauria de poder identificar un PUF d'una sèrie de PUFs aparentment iguals.

- Fiablilitat: $R = \Pi(C)$ ha de ser repetible. Al aplicar el mateix *challenge* a un mateix PUF en diferents circumstàncies, la *response* ha de ser la mateixa.

- No clonable: donada $\Pi$ es difícil de construir una funció $\Gamma \neq \Pi$ de manera que per $\forall\, C \in \chi : \Gamma \approx \Pi$ . Un PUF ha de ser difícil de copiar per tal de no comprometre la seguretat d'aquests dispositius [22].

- Imprevisible: donat un subconjunt $Q = \{(C_i, R_i = \Pi(C_i))\}$, és difícil predir si $R_c \approx \Pi(C_c)$, donada un *challenge* $C_c$ aleatoria. A partir de l'estudi d'un subconjunt de CRPs no s'ha de poder inferir la *response* del PUF a un *challenge* nou amb un temps computacional factible.

- Funció d'una direcció: donada únicament una *response* $R$, i el PUF $\Pi$, és difícil saber quin *challenge* $C$ compleix $\Pi(C) = R$.

- Evidència de manipulació: alterar el dispositiu físic que conté $\Pi$, transforma $\Pi \rightarrow \Pi'$ de manera que $\Pi(C) \neq \Pi'(C)$. Si hi ha hagut algun intent de manipulació del PUF, aquest s'ha de comportar de forma diferent per evitar atacs físics al dispositiu [23].

### 4.4.2. Paràmetres de mesura de les propietats d'un PUF

Per tal d'avaluar i comparar les propietats típiques d'un PUF integrat en un xip, Maiti *et al*. [24] proposen quantificar algunes propietats del PUF mitjançant 4 paràmetres. Aquests són: unicitat, fiabilitat, uniformitat i *bit-aliasing*.

#### 4.4.2.1. Unicitat

La unicitat d'un PUF és una mesura de la possibilitat de produir instàncies amb un comportament diferenciable. Un dels paràmetres per a mesurar la unicitat d'un PUF és la *inter-distance*. Per a un cert *challenge* es defineix la *inter-distance* entre dos instàncies diferents de PUF com la distància de Hamming en les *responses* resultants d'aplicar el mateix *challenge* a les dos instàncies del PUF. Per tal de mesurar la unicitat d'un conjunt d'instàncies d'un PUF, es defineix el següent paràmetre de unicitat:

$$Unicitat = \frac{2}{k(k-1)} \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \frac{HD(Ri, Rj)}{n} \; x \; 100 \qquad \text{(eq 4.2)}$$

On: $k$ és el nombre de PUFs avaluats; $n$ és el nombre de bits de la *response* del PUF; $HD(x, y)$ és la distància de Hamming de les paraules $x$ i $y$ ; i $R_i$ és la *response* del PUF $i$.

La *response* d'un PUF electrònic és una cadena de bits, i per aconseguir una màxima riquesa de respostes possible, cal que la *response* d'una instància es diferenciï d'una altra en la meitat dels bits. El valor ideal de la unicitat és del 50%, que assegura la màxima diferenciació, i identificació, dins d'un conjunt d'instàncies de PUF. Aquest paràmetre és també una estimació de com afecten les variacions de procés a la *response* del PUF.

#### 4.4.2.2. Fiabilitat

La fiabilitat d'un PUF és una mesura de la robustesa enfront a pertorbacions externes. La *intra-distance* és un paràmetre usualment utilitzat per mesurar com afecta el soroll a la *response* d'un PUF. Per a un cert *challenge* es defineix la *intra-distance* entre dos avaluacions d'un mateix PUF com a la distància de Hamming en les *responses,* després d'aplicar el mateix *challenge* dues vegades al mateix PUF. La fiabilitat d'un PUF es defineix amb la següent equació:

$$HD_{INTRA} = \frac{1}{m} \sum_{t=1}^{m} \frac{HD(R_i, R'_{i,t})}{n} \; x \; 100 \qquad \text{(eq 4.3)}$$

$$Fiabilitat = 100\% - HD_{INTRA} \qquad \text{(eq 4.4)}$$

On: $m$ és el nombre de CRPs avaluats; $n$ és el nombre de bits de la *response* del PUF; $HD(x, y)$ és la distància de Hamming de les paraules $x$ i $y$ ; i $R_i$és la *response* del PUF $i$.

Una fiabilitat del 100% indica una màxima robustesa enfront a pertorbacions externes, es a dir que la *response* del PUF no es veurà afectada per les pertorbacions externes.

#### 4.4.2.3. Uniformitat

La uniformitat és una mesura de la proporció de bits '0' i de bits '1' en la *response* d'un PUF. Per a mesurar la uniformitat s'utilitza el pes de Hamming per a la cadena de bits que formen la *response* d'un PUF. A continuació es mostra la mesura de la uniformitat per a una instància $i$ d'un PUF:

$$(Uniformitat)_i = \frac{1}{n}\sum_{l=1}^{n} r_{i,l}\ x\ 100 \qquad \text{(eq 4.5)}$$

On: $n$ és el nombre de bits de la *response* del PUF; i $r_{i,l}$ és el bit $l$ de la *response* del PUF $i$.

Per a assegurar que un PUF produeix *responses* totalment aleatòries i complexes de predir, la uniformitat ha de tenir una distribució gaussiana centrada al 50%.

#### 4.4.2.4. Bit-Aliasing

El *bit-aliasing* es manifesta quan varies instàncies d'un PUF produeixen un bit pràcticament idènti en les *responses* sigui pràcticament idèntic. El *bit-aliasing* es mesura utilitzant el pes de Hamming per al bit $l$ de la *response* del PUF. A continuació es mostra la mesura del *bit-aliasing* per a un bit $l$ de la response d'un conjunt d'instàncies d'un PUF:

$$(Bit - Aliasing)_l = \frac{1}{k}\sum_{i=1}^{k} r_{i,l}\ x\ 100 \qquad \text{(eq 4.6)}$$

On: $k$ és el nombre de PUFs avaluats; i $r_{i,l}$ és el bit $l$ de la *response* del PUF $i$.

L'efecte del *bit-aliasing* a la *response* dels PUFs es minimitza quan el paràmetre de l'equació (4.6) presenta una distribució gaussiana amb una mitjana propera al 50%. D'aquesta manera s'assegura una distribució uniforme de '0' i de'1' en un mateix bit de la *response* de varies instàncies PUFs.

# 5. BR-PUF: PUF basat en un anell de biestables

L'any 2011, Chen *et al*. presenten a l'article "The Bistable Ring PUF: A New Architecture for Strong Physical Unclonable Functions" [1] una arquitectura de PUF basada en anells d'inversors biestables anomenada *Bistable Ring* PUF (BR-PUF). Aquesta estructura està dissenyada per a tenir bones propietats com a PUF, millorant les característiques d'altres implementacions. Una de les principals avantatges d'aquest sistema és que amb un circuit amb pocs elements es pot generar un elevat nombre de CRPs. A més, el sistema és complex de modelitzar i parametritzar, i en conseqüència de predir les seves *responses* futures. Degut a aquestes bones característiques aquest és el PUF analitzat en aquest PFC.

La idea principal d'aquesta estructura és la creació d'un anell amb un nombre parell d'inversors. Aquest anell té un comportament biestable, es a dir, té dos estats estables diferents [1,0,1,0,...,0,1,0] o [0,1,0,1,...,1,0,1], com es mostra a la figura 5.1. Degut a que la sortida d'un inversor està connectat a l'entrada del següent, el sistema es pot estabilitzar cap a un dels dos estats estables possibles, de forma similar al cas d'una cel·la de memòria SRAM. L'estat final depèn de com afecten les variacions de procés a les propietats dels inversors.

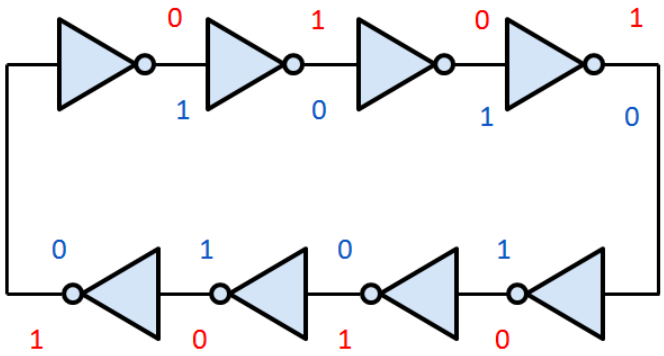


Figura 5.1: Anell biestable amb els seus estats estables

Per tal de crear un sistema amb el comportament d'un PUF a partir d'aquesta estructura cal duplicar el nombre d'inversors de tal manera que es pugui seleccionar entre parelles de dos inversors quin dels dos intervindrà en l'anell, d'acord amb el *challenge*. La selecció de l'inversor es fa amb multiplexors i demultiplexors segons s'indica a la figura 5.2. Aplicant un cert *challenge* al PUF es seleccionen els inversors que formen l'anell. L'anell tendeix cap a un dels dos estats biestables possibles, donant lloc a la *response*.

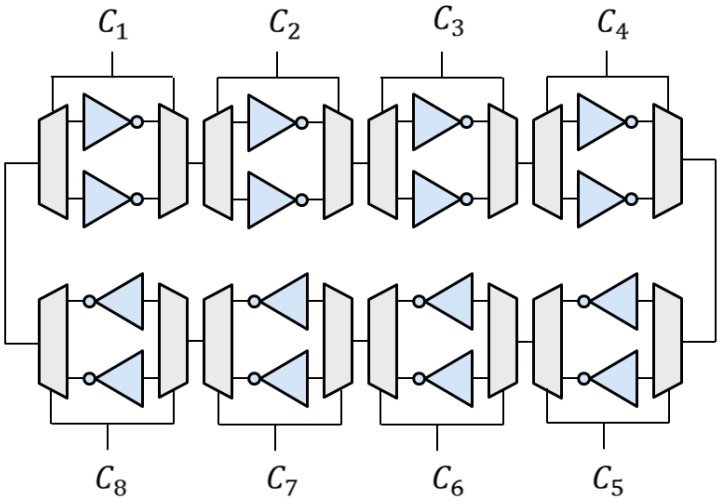


Figura 5.2: Arquitectura del BR-PUF

Una primera definició de l'obtenció de la *response* del BR-PUF consisteix en observar quin és l'estat final al que tendeix l'anell biestable. Si l'anell tendeix cap al estat [1,0,1,0,...,0,1,0] es considera que la *response* és '1', si el sistema s'estabilitza en [0,1,0,1,...,1,0,1] es considera que la *response* del PUF és '0'.

Com ja s'ha comentat aquest sistema presenta un elevat nombre de CRPs. Per a $N$ bits del *challenge* es poden arribar a generar $2^N$ CRPs. Cal destacar que per a $N$ bits del *challenge*, el BR-PUF necessita $2N$ inversors, $N$ multiplexors i $N$ demultiplexors. A més, al aplicar un *challenge*, es forma un anell biestable de $N$ inversors. El conjunt de 2 inversors multiplexor i demultiplexor rep el nom d'etapa del BR-PUF.

En l'anàlisi d'aquest sistema s'utilitzen dos metodologies diferents. Per una banda es realitzen simulacions SPICE del circuit electrònic amb el programari Cadence Virtuoso. Per altra banda es crea un model matemàtic del sistema i s'analitza utilitzant les llibreries Scipy i Numpy del llenguatge de programació Python. S'utilitzen dues metodologies diferents d'anàlisi perquè, tot i que el programari Cadence Virtuoso proporciona unes simulacions molt acurades, també tenen un temps d'execució molt elevat, dificultant la caracterització i estudi del circuit. Amb la utilització d'un model matemàtic es simula el comportament del BR-PUF amb més rapidesa, generant suficients dades per a mesurar les propietats del PUF (per exemple la unicitat o la uniformitat).

## 5.1. Estudi del BR-PUF amb el programari Cadence Virtuoso

En primer lloc s'ha implementat un anell biestable de 64 inversors mitjançant el programari Cadence Virtuoso. Els inversors s'han realitzat amb la tecnologia de 65nm cmos065 i s'han dimensionat de manera que el retard de pujada sigui igual al retard de baixada. Amb l'ajut

d'aquest programa, es simula el comportament transitori des d'un estat inicial inestable (en aquest cas es fixa que tots els inversors generin un '0' a l'estat inicial) fins a un estat estable. A la figura 5.3 es mostra com evoluciona la sortida d'un inversor durant 100ns.

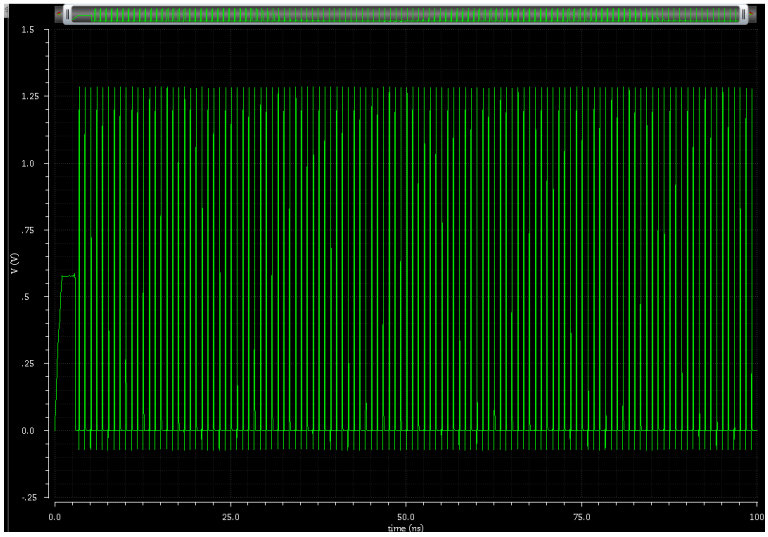


Figura 5.3: Evolució de la sortida d'un inversor en un anell de 64 inversors durant 100ns

S'observa que en aquesta simulació es produeixen una sèrie d'oscil·lacions que provoquen que el transitori no s'hagi acabat al finalitzar la simulació (100ns). Per tant amb 100ns no n'hi hauria prou per a obtenir la *response* del PUF.

El programari permet també simular les variacions del procés mitjançant simulacions Monte Carlo, donant lloc als resultats estadístics corresponents. Quan es simulen 200 instàncies de l'anell biestable en aquestes condicions s'obtenen les gràfiques que es mostren a la figura 5.4.

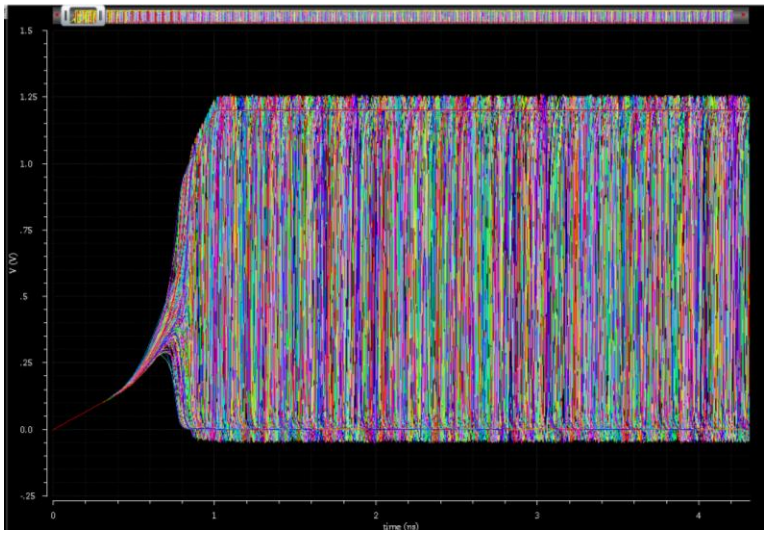


Figura 5.4: Evolució de la sortida d'un inversor en la simulació de 200 anells biestables

Es veu que en tots els casos es generen oscil·lacions durant el transitori. Això té l'inconvenient de que resulta lent obtenir un elevat nombre de dades mitjançant simulacions SPICE. Cal simular el circuit durant un temps suficientment gran per tal d'assegurar que aquest ha arribat a un estat estable. A la pràctica, les oscil·lacions transitòries provoquen que obtenir un nombre significatiu de CRPs sigui bastant lent ja que s'ha d'esperar a que l'anell s'estabilitzi.

## 5.2. Estudi del BR-PUF mitjançant un model matemàtic

L'obtenció d'un nombre elevat de CRPs per a la caracterització del BR-PUF resulta un procés lent mitjançant l'ús de simulacions SPICE. Aquest fet es degut a que el model SPICE simula el comportament d'un circuit electrònic tenint en compte tots els paràmetres dels transistors que el formen i calculant totes les tensions i corrents involucrades en el circuit. En aquest PFC s'ha estudiat i simulat un model simplificat del circuit que caracteritza el BR-PUF amb un nombre inferior de paràmetres permetent reduir significativament el temps de simulació.

### 5.2.1. Model matemàtic del BR-PUF

Per tal de simular el comportament dels anells biestables de forma ràpida, s'ha utilitzat el model de gran senyal de l'inversor [25] que es mostra a la figura 5.5. Per facilitar l'estudi matemàtic del model, s'ha suposat que l'inversor està alimentat per les tensions $v_{dd}/2$ (nivell alt) i $-v_{dd}/2$ (nivell baix). En aquest model, l'inversor es troba en el punt metastable quan la tensió d'entrada $v_{i-1}$ és 0 V.

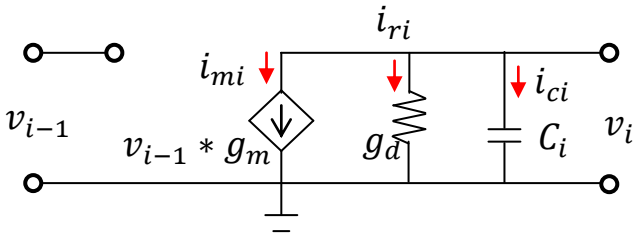


Figura 5.5: Model de gran senyal de l'inversor

Les intensitats que circulen pel model del inversor es defineixen de la forma següents:

$$i_{mi} = g_m v_{i-1} \quad \text{(eq 5.1)}$$

$$i_{ri} = g_d v_i + (g_m - g_d)\left(\frac{2v_i}{v_{dd}}\right)^3 \frac{v_{dd}}{2} \quad \text{(eq 5.2)}$$

$$i_{Ci} = C_i \dot{v}_i \quad \text{(eq 5.3)}$$

L'únic paràmetre que s'ha suposat que varia significativament entre dues instàncies d'inversors és la capacitat d'aquest ($C_i$).

Cal notar que en la equació 5.2, que representa la intensitat que circula per la conductància, hi apareixen dos termes. El primer terme $g_d v_i$ representa la intensitat que circularia per una conductància de valor $g_d$. El segon terme serveix per simular la saturació en la sortida de l'inversor quan se li aplica una tensió en nivell alt ($v_{dd}/2$) o de nivell baix (-$v_{dd}$/2).

Quan l'inversor rep a l'entrada $-v_{dd}/2$, la font de tensió genera un corrent de $-g_m v_{dd}/2$. En aquest cas la sortida del inversor és $v_{dd}/2$. A partir de la equació 5.2, s'obté que per la conductància hi circula una intensitat de $g_m v_{dd}$/2. Per tant, la intensitat que circula pel condensador és 0 (equació 5.3). Al no circular intensitat pel condensador, aquest no es carrega ni descarrega, per tant l'inversor manté una tensió de $v_{dd}/2$ quan a l'entrada té una tensió de $-v_{dd}/2$. De forma anàloga a l'aplicar una tensió de $v_{dd}/2$ a l'inversor, aquest genera $-v_{dd}/2$ a la seva sortida.

Els inversors realitzats amb tecnologia CMOS absorbeixen una petita intensitat per la entrada degut a la seva capacitat paràsita. Per menysprear aquest fet, el model concentra aquesta intensitat a $C_i$. D'aquesta manera es pot assumir que la intensitat de sortida de l'inversor sigui 0. Per tant, l'equació bàsica del model matemàtic del BR-PUF és:

$$i_{ri} + i_{mi} + i_{Ci} = 0 \qquad \text{(eq 5.4)}$$

S'ha aplicat el següent canvi de variables a les definicions de la intensitat per simplificar l'equació bàsica dels inversors 5.4 a l'equació 5.8:

$$x_i = \frac{2v_i}{v_{dd}} \qquad \text{(eq 5.5)}$$

$$\rho = \frac{g_d}{g_m} \qquad \text{(eq 5.6)}$$

$$\omega_i = \frac{g_m}{C_i} \qquad \text{(eq 5.7)}$$

$$\dot{x}_i = -\omega_i\left(x_{i-1} + \rho x_i + (1-\rho)x_i^3\right) \qquad \text{(eq 5.8)}$$

Amb aquest canvi de variables, la variable $x_i$ representa l'estat d'un inversor. El seu valor és '-1' si la tensió de sortida de l'inversor és $-v_{dd}/2$. En canvi, $x_i$ val '1' si la sortida de l'inversor és $v_{dd}/2$.

Considerant un anell inversor de $N$ etapes, s'obté el següent sistema d'equacions diferencials (EDOs):

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \vdots \\ \dot{x}_{N-1} \\ \dot{x}_N \end{pmatrix} = \begin{pmatrix} -\omega_i(x_N + \rho x_1 + (1-\rho)x_1^3) \\ -\omega_i(x_1 + \rho x_2 + (1-\rho)x_2^3) \\ -\omega_i(x_2 + \rho x_3 + (1-\rho)x_3^3) \\ -\omega_i(x_3 + \rho x_4 + (1-\rho)x_4^3) \\ \vdots \\ -\omega_i(x_{N-2} + \rho x_{N-1} + (1-\rho)x_{N-1}^3) \\ -\omega_i(x_{N-1} + \rho x_N + (1-\rho)x_N^3) \end{pmatrix} \quad \text{(eq 5.9)}$$

#### 5.2.1.1. Extracció dels paràmetres del model matemàtic de l'inversor

Per tal d'ajustar el model de l'inversor al comportament real, els paràmetres $g_m$, $g_d$ i $C_i$ s'han extret a partir d'un conjunt de simulacions realitzades amb la tecnologia de cmos065. La parametrització realitzada consisteix en generar un conjunt de simulacions sobre l'inversor a parametritzar. A continuació es mostren els passos que s'han seguit per parametritzar un inversor de la tecnologia cmos065:

##### 5.2.1.1.1 Obtenció del paràmetre $g_m$

Per a obtenir el paràmetre $g_m$, s'ha simulat, utilitzant el programari Cadence, un inversor tal i com es mostra en la figura 5.6. Tant a l'entrada com a la sortida de l'inversor s'imposa una tensió en contínua $v_M$ igual a la tensió corresponent al punt metaestable de l'inversor. Cal destacar que $v_M$ en el model matemàtic equival a una tensió nul·la (0 V), en canvi en un inversor alimentat entre $v_{dd}$ i terra (0 V), aquest nivell és de $v_{dd}/2$. A més, s'afegeix una senyal alterna de poca amplitud a l'entrada de l'inversor. A partir de la equació 5.2 es dedueix que la intensitat teòrica que circula per la conductància és 0 si l'amplitud de tensió alterna aplicada a l'inversor és menyspreable en front a la tensió d'alimentació. A més, com que la tensió de sortida és constant, el condensador tampoc aporta corrent a la sortida de l'inversor. Per tant, tota la intensitat que aporta l'inversor a la sortida es deguda a la font de corrent del model. Tenint en compte la tensió alterna a l'entrada de l'inversor i la corrent de sortida de l'inversor, es pot calcular el paràmetre $g_m$ a partir de la equació 5.10.

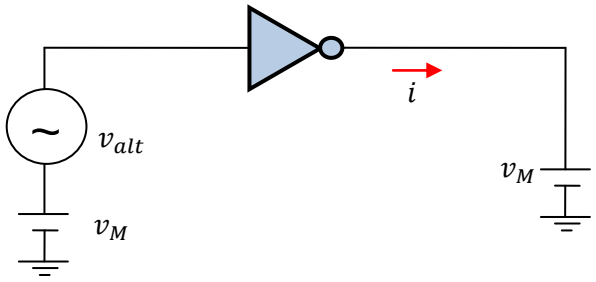


Figura 5.6 Obtenció del paràmetre $g_m$

$$g_m = \frac{\max(i)}{\max(v_{alt})} \qquad \text{(eq 5.10)}$$

Per a un inversor de la tecnologia cmos065, s'ha obingut el següent paràmetre: $g_m = 5{,}2 \cdot 10^{-4}\Omega^{-1}$.

#### 5.2.1.1.2 Obtenció del paràmetre $g_d$

Per obtenir el paràmetre $g_d$ s'ha simulat un inversor utilitzant, el programari Cadence, tal i com es mostra a la figura 5.7. El fet d'aplicar una tensió de $v_M$ (0 V en el model matemàtic) a l'entrada de l'inversor provoca que la tensió de corrent teòrica del model no aporti corrent a la sortida de l'inversor. A més, si la tensió de sortida és constant, el condensador tampoc aporta corrent. Per tant tota la corrent que surt de l'inversor es deguda a la conductància del model. Imposant una tensió de sortida constant i igual a $v_M + \Delta v$ (recordar que per al model matemàtic $v_M = 0$ ), l'equació 5.2 es transforma en $i_r = g_d \Delta v$ ja que el segon terme de la equació es pot menysprear si $\Delta v$ és suficientment petit. Per tant, finalment s'ha arribat a l'equació 5.11.

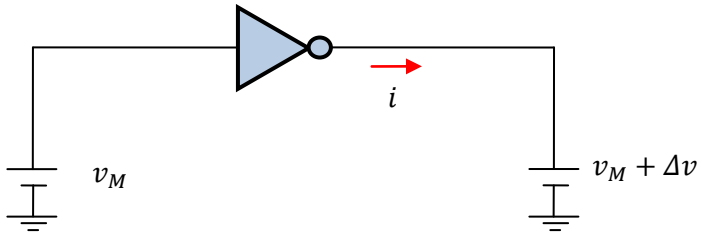


Figura 5.7 Obtenció del paràmetre $g_d$

$$g_d = \frac{i}{\Delta v} \qquad \text{(eq 5.11)}$$

Per a un inversor de la tecnologia cmos065, s'ha obtingut el següent paràmetre: $g_d = 5{,}86 \cdot 10^{-5}\Omega^{-1}$.

#### 5.2.1.1.3 Obtenció del paràmetre $C_i$

Per obtenir el paràmetre $C_i$ s'ha simulat un inversor utilitzant, el programari Cadence, tal i com es mostra a la figura 5.8. S'han aplicat polsos de tensió a les entrades de dos cadenes d'inversors. Ajustant la capacitat del condensador per tal que l'evolució transitòria de la tensió a la sortida del segons inversors en les dos branques, es pot aproximar el paràmetre $C_i$ del model.

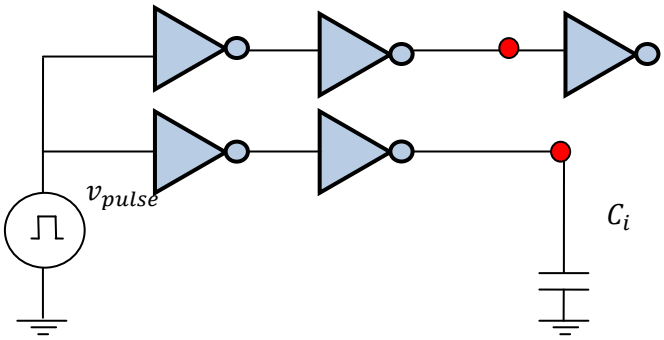


Figura 5.8 Obtenció del paràmetre $C_i$

Per a un inversor de la tecnologia cmos065, s'ha obtingut el següent paràmetre: $C = 5fF$.

#### 5.2.1.2. Simulació del model matemàtic del BR-PUF

Mitjançant la creació d'un entorn de simulació amb el llenguatge de programació Python, i les seves llibreries científiques Scipy i Numpy, s'ha simulat el comportament d'un anell d'inversors. Aquest entorn permet integrar el sistema d'EDOs (equació 5.9) generant el transitori dels anells d'inversors des d'un estat inicial a un estat estable. A la figura 5.9 es mostra l'evolució de l'estat d'un inversor en un anell de 64 inversors al llarg de 100ns.

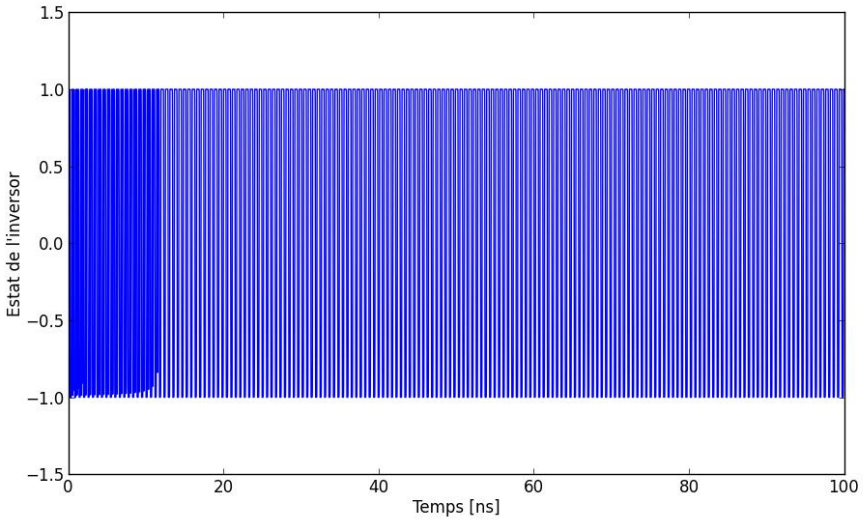


Figura 5.9: Evolució de l'estat d'un inversor en un anell de 64 inversors

Comparant aquest gràfic amb el de la figura 5.3 s'observa que les simulacions realitzades amb models SPICE tenen resultats molt similars a les simulacions basades en el model matemàtic del inversor. Cal destacar que en aquest gràfic l'estat de l'inversor varia entre '-1' i '1' degut a com s'ha definit el model matemàtic.

### 5.2.2. Anàlisi de la *response* del model matemàtic del BR-PUF

Per tal de poder avaluar la *response* del BR-PUF abans de que aquest s'hagi estabilitzat, es pot realitzar una obtenció de la *response* del BR-PUF alternativa a la descrita a l'inici del capítol 5 [3]. Tenint en compte que l'estat de l'inversor, pot variar entre -1 i 1, es defineixen els següents estats estables d'un anell d'inversors biestable : $s^+ = [1, -1, \dots, 1, -1]$ i $s^+ = [-1,1, \dots, -1,1]$. Per a obtenir la *response* del PUF, primerament cal emmagatzemar l'estat dels inversors de l'anell, un cert temps després de posar en funcionament el circuit ($t_s$), en un vector $x(t_s)$. La *response* del BR-PUF s'obté a l'aplicar la funció signe (definida a l'equació 5.12) al producte escalar entre $x(t_s)$ i $s^+$. L'equació 5.13 mostra com realitzar aquest càlcul. A partir d'aquest punt la *response* del BR-PUF es calcularà utilitzant aquesta metodologia.

$$signum\,(x) = \begin{cases} 1 & si\ x > 0 \\ 0 & si\ x = 0 \\ -1 & si\ x < 0 \end{cases} \qquad (\text{eq } 5.11)$$

$$r = signum\left(\sum_{n=0}^{N-1} s_n^+ \cdot x_n(t_s)\right) \qquad (\text{eq } 5.12)$$

Per tal de caracteritzar el BR-PUF s'han simulat 50 BR-PUFs de 32 inversors (que, per tant, tenen 32 bits de *challenge*) als quals se'ls hi ha aplicat els mateixos 128 *challenges*. Tot i que fins ara s'han simulat BR-PUFs de 64 inversors, s'ha decidit caracteritzar anells biestables de 32 inversors per a poder generar suficients dades en un temps raonable. El primer que cal destacar de les simulacions és que els anells biestables tendeixen a oscil·lar durant un temps considerable, fent que en alguns casos la simulació s'aturi abans de que s'hagi estabilitzat l'anell d'inversors. Aquest fenomen s'observa en la figura 5.10 com el darrer segment vertical de la gràfica. El segment es vertical perquè tots els PUFs no s'han estabilitzat al finalitzar la simulació. S'observa que aproximadament un 45% dels anells simulats seguirien oscil·lant si la simulació continués. Com ja s'ha comentat aquest fet pot enrederir la obtenció de CRPs en una implementació física del BR-PUF.

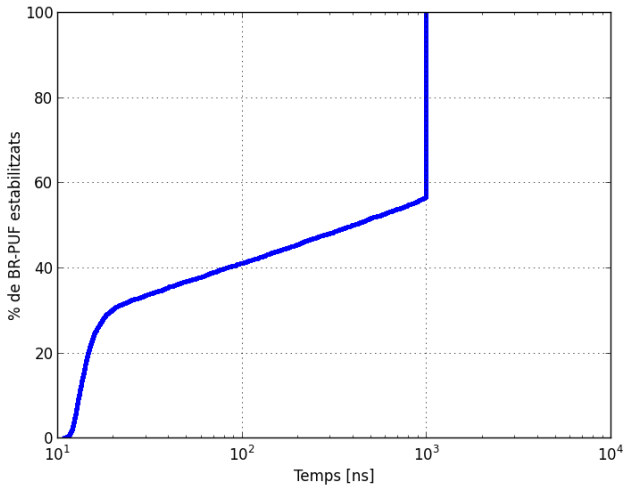


Figura 5.10: Funció de distribució acumulada del temps del temps d'estabilització dels anells d'inversors. El segment vertical representa les instàncies que en acabar la simulació seguirien oscil·lant

Mitjançant les equacions 4.2, 4.5, i 4.6 s'ha analitzat la *response* obtinguda en els BR-PUFs simulats. Cal notar que la fiabilitat del PUF no s'ha calculat ja que el model no inclou la possibilitat d'afegir variacions generades per pertorbacions externes.

#### 5.2.2.1. Unicitat

La figura 5.11 mostra un histograma de les *inter-distances* obtingudes entre totes les *responses* dels 50 BR-PUFs.

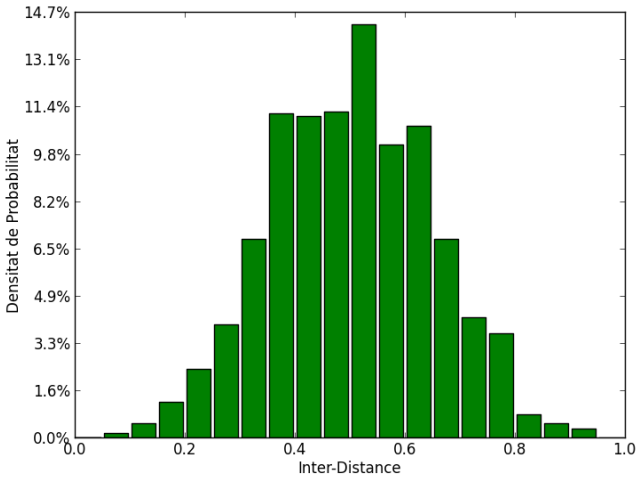


Figura 5.11: Histograma de les *inter-distances* entre les *responses* dels BR-PUFs simulats amb 32 bits de *challenge*

Al gràfic s'aprecia que les *inter-distances* tenen una distribució gaussiana centrada al 0.5. A més la unicitat val 50,25%, un valor molt pròxim al valor òptim d'aquest paràmetre (50%). Per tant aquesta tipologia de PUF permet crear instancies diferenciables entre elles.

#### 5.2.2.2. Uniformitat

La figura 5.12 mostra un histograma de les uniformitats obtingudes en els 50 BR-PUFs.

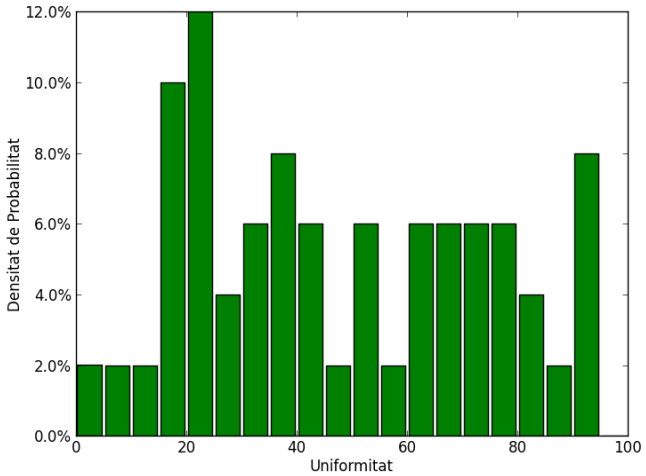


Figura 5.12: Histograma de les uniformitats dels BR-PUFs simulats amb 32 bits de *challenge*

En aquesta gràfica s'observa que la uniformitat dels PUFs té una distribució completament aleatòria. Aquest fet provoca que en algunes instàncies el comportament del PUF és fàcil de predir, sobretot en els casos que presenten una uniformitat del 0% o 100%, ja que la *response* donaria sempre el mateix valor.

#### 5.2.2.3. Bit-aliasing

La figura 5.13 mostra un histograma del *bit-aliasing* obtingut en els 32 *challenge* dels BR-PUFs simulats.

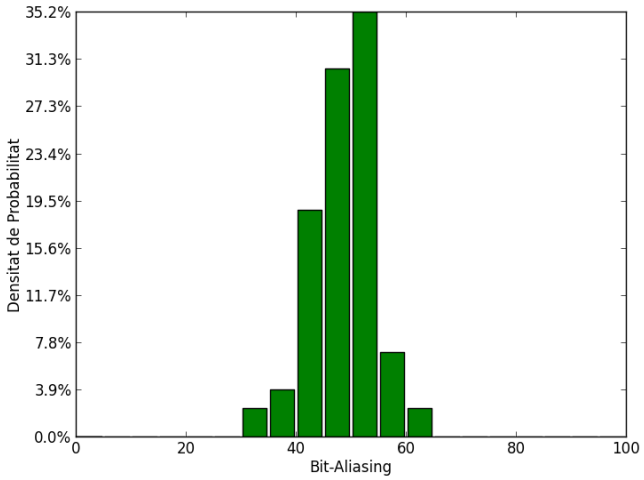


Figura 5.13 Histograma del *bit-aliasing* dels BR-PUFs simulats amb 32 bits de *challenge*

El *bit-aliasing* té una mitjana del 48,02%, un valor molt pròxim al valor òptim d'aquest paràmetre (50%). Per tant aquesta tipologia de PUF no presenta el problema de *bit-aliasing*.

A excepció de la uniformitat aquesta estructura presenta molt bones propietats per a funcionar com a PUF. A més el fet de que sigui complex de modelar i de predir, permet crear un dispositiu amb més seguretat que altres tipologies com pot ser l'*arbiter* PUF.

## 5.3. Modificacions del BR-PUF

En l'apartat anterior s'ha observat que el BR-PUF presenta dos grans problemes. Per una banda, aquest sistema tendeix a oscil·lar, complicant i endarrerint l'obtenció d'un elevat nombre de CRPs. A més, analitzant les *responses* obtingudes, s'ha observat que aquesta tipologia presenta una distribució de la uniformitat aleatòria. Aquest fet està provocat perquè en algunes instàncies del PUF, la *response* tendeix cap al mateix valor amb una elevada probabilitat.

A continuació es presenten dues modificacions que es poden realitzar en el BR-PUF Per a solucionar aquests problemes per tal de millorar el seu comportament.

### 5.3.1. IOCF

Per tal d'eliminar les oscil·lacions transitòries que pateix el BR-PUF, s'ha dissenyat un filtre anomenat *Interleaved Oscillation Canceller Filter* (IOCF) [8]. Aquest filtre consisteix en connectar un condensador entre dos inversors consecutius de l'anell com es mostra en la figura 5.14.

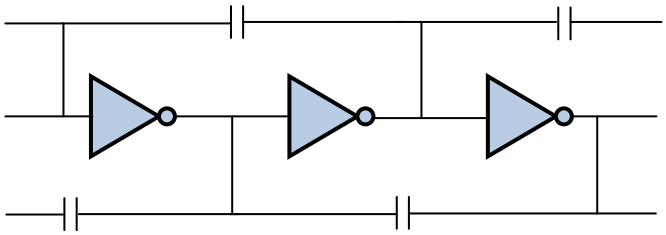

Figura 5.14: Estructura del filtre IOCF

Aquesta estructura s'estudiarà amb detall la secció 6.

### 5.3.2. TBR-PUF

El problema de la uniformitat sorgeix a causa de la possibilitat de que durant la construcció de l'anell aparegui un inversor que domini sobre la resta d'inversors de l'anell. Per tant, sempre que un *challenge* seleccioni aquest inversor, el BR-PUF donarà sempre la mateixa *response.* Per tal de solucionar aquest problema s'ha presentat una estructura anomenada *Twisted Bistable Ring* PUF (TBR-PUF) [3]. Tal i com es mostra en la figura 5.15, en aquesta estructura tots els inversors formen part de l'anell, i el *challenge* selecciona quin inversor de l'etapa forma part de la cadena superior de l'anell i quin de forma part de la cadena inferior de l'anell o dit d'una altra manera, el *challenge* reordena les posicions que ocupen els inversors dins de l'anell.

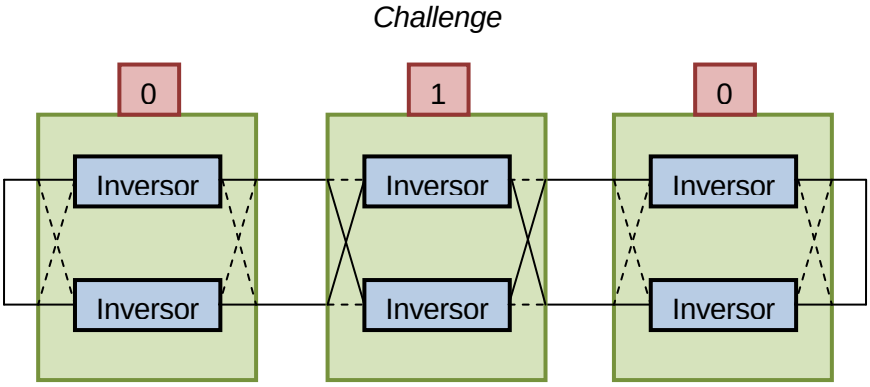


Figura 5.15: Estructura del TBR-PUF

Aquesta estructura s'estudiarà amb detall al capítol 7.

# 6. IOCF: Filtre cancel·lador d'oscil·lacions

Mitjançant l'ús del filtre IOCF presentat a l'apartat 5.3.1 s'arriben a cancel·lar les oscil·lacions que es produeixen en el transitori del BR-PUF. El principi de funcionament d'aquest filtre es basa en que la impedància d'un condensador és funció de la freqüència: ($Z = 1/j\omega C_f$). Quan l'anell està oscil·lant a elevada freqüència ($\omega \gg 0$), la impedància de la capacitat tendeix cap a 0. Es a dir, en aquest cas, el condensador tendeix a comporta-se com un curtcircuit entre l'entrada del primer i la sortida del segon inversor. D'aquesta manera s'aconsegueix que dues etapes consecutives de l'anell es comportin com una cel·la SRAM, que és intrínsecament estable (una cel·la SRAM està formada per dos inversors realimentats). En canvi quan el sistema es troba en un estat estable ($\omega \simeq 0$), la impedància del condensador tendeix a infinit fent que es comporti com un interruptor obert. En aquest cas, l'estructura completa es comporta com l'anell d'inversors original. La figura 6.1 il·lustra el principi de funcionament del filtre.

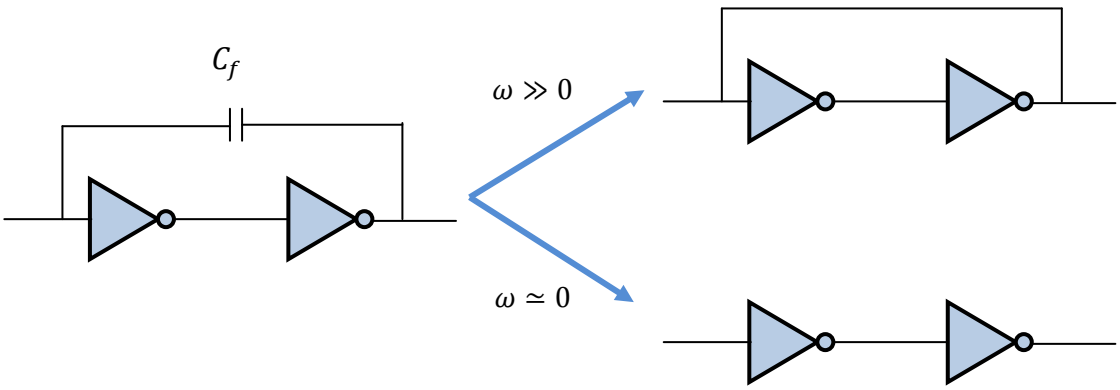


Figura 6.1: Principi de funcionament del filtre IOCF

Per tal d'estudiar el comportament del filtre i de com aquest afecta a les propietats del PUF, es segueix la mateixa metodologia emprada en estudiar el BR-PUF. Per una banda es realitzen simulacions SPICE del circuit electrònic que generen un nombre de dades suficients per a caracteritzar el BR-PUF amb IOCF i s'amplia el model matemàtic del PUF incloent-hi el filtre IOCF. Després es realitzen simulacions del model matemàtic amb l'ajuda del llenguatge de programació Python.

## 6.1. Estudi del filtre IOCF amb el programari Cadence Virtuoso

S'han implementat anells biestables de 64 inversors amb filtres IOCF de diferents capacitats, a fi d'observar el comportament del filtre IOCF. Com en l'apartat 5.1 s'han emprat inversors equilibrats de la tecnologia cmos065. Les següents figures mostren l'evolució de 200 simulacions Monte Carlo per a diferents valors de la capacitat $C_f$ del filtre IOCF.

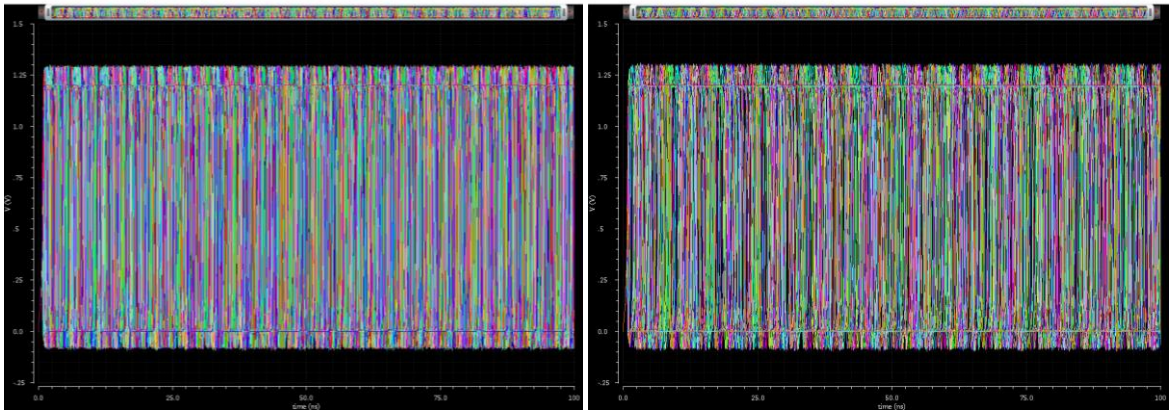

Figura 6.2: Transitori de 200 anells biestables amb filtre IOCF de $C_f = 5fF$ i de $C_f = 10fF$

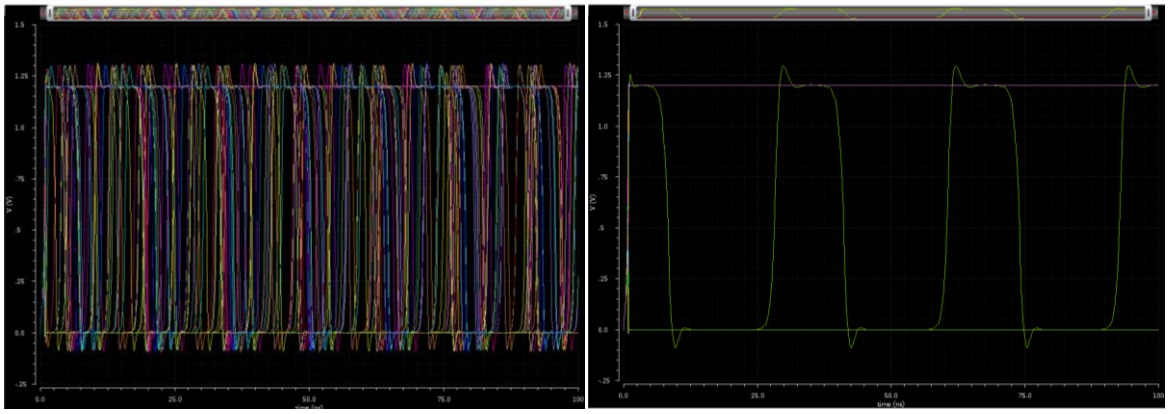

Figura 6.3: Transitori de 200 anells biestables amb filtre IOCF de $C_f = 20fF$ i de $C_f = 50fF$

En aquestes figures s'observa que a mesura que la capacitat del condensador augmenta, les oscil·lacions tendeixen a fer-se més lentes, i, a partir de cert punt desapareixen. Per a una capacitat de $C_f = 50fF$ (figura 6.3) s'observa que nomes en un cas, l'anell comença a oscil·lar. Es a dir que amb una capacitat de $50fF$ les oscil·lacions es cancel·len en un 99,5%. Per tant, amb un condensador amb una capacitat suficientment gran es poden cancel·lar totes les oscil·lacions.

## 6.2. Estudi del filtre IOCF mitjançant el model matemàtic

Tot i que els resultats obtinguts en l'apartat anterior són molt positius, cal comprovar que el fet d'afegir el filtre no afecta a les característiques del PUF. Per a obtenir un elevat nombre de dades s'utilitza el model matemàtic ampliat, descrit en al principi de la secció 6.

### 6.2.1. Model matemàtic del BR-PUF amb IOCF

Per tal d'incloure el filtre al model matemàtic, cal tenir en compte que a la sortida de cada inversor, hi actuen dos condensadors, tal i com mostra la figura 6.4. La intensitat que circula per cada condensador ve definida per la equació 6.1. Tenint en compte aquests dos fets, la intensitat que surt de l'inversor segueix l'equació 6.2. Així doncs, l'equació bàsica del model 5.4 es converteix en 6.3.

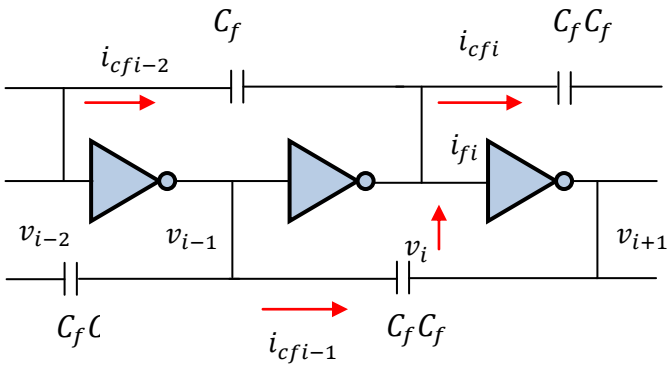


Figura 6.4: Definició de les intensitats que circulen pels condensadors

$$i_{Cfi} = C_f\left(\frac{d}{dt}(v_{i+2} - v_i)\right) \quad \text{(eq 6.1)}$$

$$i_{fi} = i_{Cfi} - i_{Cfi-2} = C_f\left(\frac{d}{dt}(v_{i-2} - v_i) + \frac{d}{dt}(v_{i+2} - v_i)\right) \quad \text{(eq 6.2)}$$

$$i_{Cfi} = i_{ri} + i_{mi} + i_{Ci} \quad \text{(eq 6.3)}$$

El canvi de variables descrit per les equacions 6.4 a 6.7 simplifica significativament el model. Tenint en compte la definició de la funció 6.8, l'equació bàsica del model es transforma en l'equació 6.9:

$$x_i = \frac{2v_i}{v_{dd}} \quad \text{(eq 6.4)}$$

$$\rho = \frac{g_d}{g_m} \quad \text{(eq 6.5)}$$

$$\omega_i = \frac{g_m}{C_i + 2C_f} \quad \text{(eq 6.6)}$$

$$\alpha_i = \frac{C_f}{C_i + 2C_f} \qquad \text{(eq 6.7)}$$

$$f_i = -\omega_i\left(x_{i-1} + \rho x_i + (1-\rho)x_i^3\right) \qquad \text{(eq 6.8)}$$

$$\dot{x}_i - \alpha_i \dot{x}_{i-2} - \alpha_i \dot{x}_{i+2} = f_i \qquad \text{(eq 6.9)}$$

Cal notar que l'equació 6.9 és una equació diferencial implícita ja que hi apareix les derivadaes temporal de 3 variables estat de l'inversor. Considerant un anell biestable de N etapes s'obté el sistema de equacions descrit a l'equació 6.10.

$$\begin{pmatrix} 1 & 0 & -\alpha_1 & 0 & \cdots & -\alpha_1 & 0 \\ 0 & 1 & 0 & -\alpha_2 & \ldots & 0 & -\alpha_2 \\ -\alpha_3 & 0 & 1 & 0 & \ldots & 0 & 0 \\ 0 & -\alpha_4 & 0 & 1 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ -\alpha_{N-1} & 0 & 0 & 0 & \ldots & 1 & 0 \\ 0 & -\alpha_N & 0 & 0 & \cdots & 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \vdots \\ \dot{x}_{N-1} \\ \dot{x}_N \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ \vdots \\ f_{N-1} \\ f_N \end{pmatrix} \qquad \text{(eq 6.10)}$$

Aquest sistema no és un sistema d'equacions diferencials tal i com està descrit a l'equació 6.10. Cal modificar-lo per a poder analitzar i simular-lo. Mitjançant la definició de la matriu $A$ descrita a l'equació 6.11, el sistema d'EDOs de l'anell biestable queda representat pel sistema descrit a l'equació 6.12.

$$A = \begin{pmatrix} 1 & 0 & -\alpha_1 & 0 & \cdots & -\alpha_1 & 0 \\ 0 & 1 & 0 & -\alpha_2 & \ldots & 0 & -\alpha_2 \\ -\alpha_3 & 0 & 1 & 0 & \ldots & 0 & 0 \\ 0 & -\alpha_4 & 0 & 1 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ -\alpha_{N-1} & 0 & 0 & 0 & \ldots & 1 & 0 \\ 0 & -\alpha_N & 0 & 0 & \cdots & 0 & 1 \end{pmatrix} \qquad \text{(eq 6.11)}$$

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \vdots \\ \dot{x}_N \end{pmatrix} = A^{-1} \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_N \end{pmatrix} \qquad \text{(eq 6.12)}$$

#### 6.2.1.1. Anàlisi del sistema d'equacions diferencials ordinàries del BR-PUF amb IOCF al voltant del punt metaestable

El sistema d'EDOs del BR-PUF permet estudiar el comportament de l'anell biestable al voltant del punt metaestable dels inversors. Aquest punt es produeix quan s'aplica una tensió de $v_M$ a la entrada de l'inversor, i, per tant, l'inversor també genera una tensió de sortida $v_M$. Aquesta situació queda representada al model matemàtic quan el vector d'estats

dels inversors és: $x = (0,0,\dots,0)^T$. Com que el sistema d'equacions no és lineal, cal linealitzar-lo. Per fer-ho s'ha d'obtenir la matriu jacobiana del sistema definida per la equació 6.13. Considerant la matriu $A^{-1}$ com una matriu en termes genèrics (equació 6.14), la derivada temporal de l'estat d'un inversor té la forma de l'equació 6.15. L'equació 6.16 mostra en termes genèrics el valor d'un terme del jacobià del sistema.

$$J = \begin{pmatrix} \frac{d\dot{x}_1}{dx_1} & \cdots & \frac{d\dot{x}_1}{dx_N} \\ \vdots & \ddots & \vdots \\ \frac{d\dot{x}_N}{dx_1} & \cdots & \frac{d\dot{x}_N}{dx_N} \end{pmatrix} \qquad \text{(eq 6.13)}$$

$$A^{-1} = \begin{pmatrix} a_{11} & \cdots & a_{1N} \\ \vdots & \ddots & \vdots \\ a_{N1} & \cdots & a_{NN} \end{pmatrix} \qquad \text{(eq 6.14)}$$

$$\dot{x}_i = a_{i1}f_1 + a_{i2}f_2 + \cdots + a_{iN}f_N \qquad \text{(eq 6.15)}$$

$$\frac{d\dot{x}_i}{dx_j} = a_{i1}\frac{df_1}{dx_j} + a_{i2}\frac{df_2}{dx_j} + \cdots + a_{iN}\frac{df_N}{dx_j} \qquad \text{(eq 6.16)}$$

Degut a que les úniques funcions $f_i$ que depenen de l'estat $x_i$ són les funcions $f_i$ i $f_{i+1}$ (equació 6.8) les derivades $\frac{df_i}{dx_j}$ valen 0 si $i \neq \{j, j+1\}$. Tenint en compte aquest fet, els termes de la matriu jacobiana tenen la següent expressió:

$$j_{ij} = \frac{d\dot{x}_i}{dx_j} = a_{ij}\left[-\omega_j\left(\rho + (1-\rho)3x_j^2\right)\right] + a_{ij+1}\left[-\omega_{j+1}\right] \qquad \text{(eq 6.17)}$$

La matriu del sistema linealitzat s'obté al substituir el punt metaestable ($x = (0,0,\cdots,0)^T$) a la matriu jacobiana. Obtenint els VAPs del sistema linealitzat, s'ha estudiat el comportament del sistema al voltant del punt metaestable. La figura 6.5 mostra el mapa de VAPs d'un anell biestable de 64 inversors sense filtre IOCF. Cal notar que hi apareixen 64 VAPs ja que per cada variable d'estat (es a dir, per a cada inversor) el sistema té un VAP.

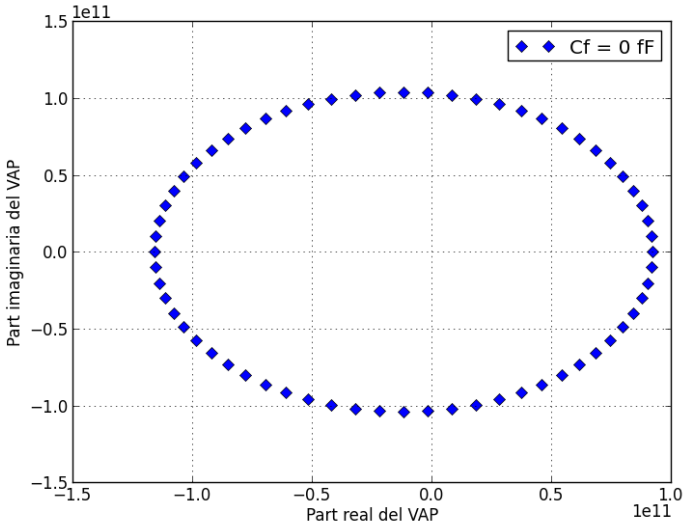


Figura 6.5: Mapa de VAPs d'un anell biestable sense filtre IOCF

Els VAPs del sistema es poden classificar en tres categories:

1. VAPs estables: VAPs amb part real negativa. Aquesta categoria de VAP no afecta gaire sobre el comportament de l'anell biestable ja que en aquest també hi apareixen VAPs inestables que dominen sobre els estables.

2. VAP inestable no oscil·lant: VAP sense part imaginària i part real positiva. Aquest VAP és l'encarregat de que l'anell biestable tendeixi a sortir del punt metaestable per a acabar en un dels dos possibles estats estables.

3. VAPs inestables oscil·lants: VAPs amb part imaginària i part real positives. Aquesta categoria de VAPs és la causant de les oscil·lacions transitòries de l'anell biestable.

La figura 6.6 mostra el mapa de VAPs obtingut a l'aplicar varis filtres IOCF de diferents capacitats a l'anell biestable de 64 inversors. La figura 6.7 mostra una ampliació de la zona inestable del mapa de VAPs, S'observa que el VAP inestable no oscil·lant es manté en el mateix punt, però que els VAPs inestables oscil·lants tendeixen a apropar-se cap a l'eix vertical. Aquest fet provoca que el VAP inestable no oscil·lant domini sobre la resta de VAPs ja que la part real d'aquest VAP és més gran que la part real de la resta dels VAPs. Per aquest motiu, s'aconsegueix cancel·lar en gran mesura les oscil·lacions transitòries de l'anell biestable.

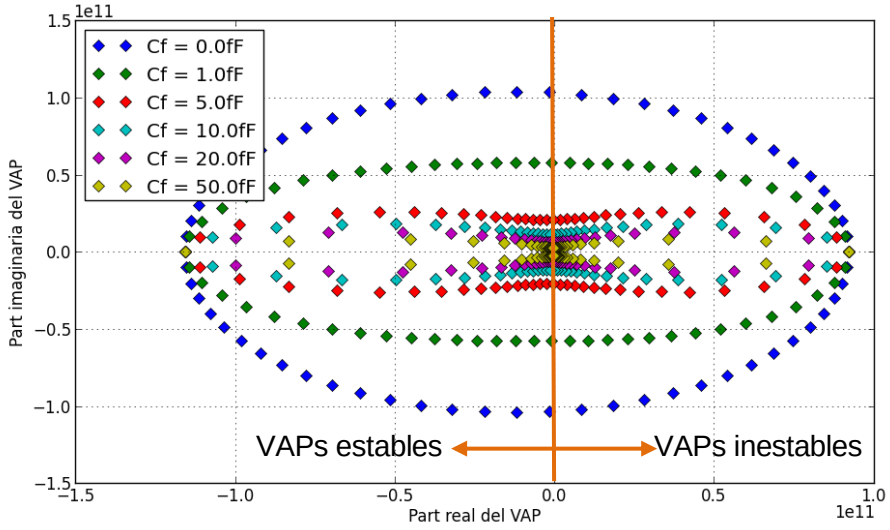


Figura 6.6: Mapa de VAPs de varis anells biestables de 64 inversors amb filtres IOCF de diferent capacitat.

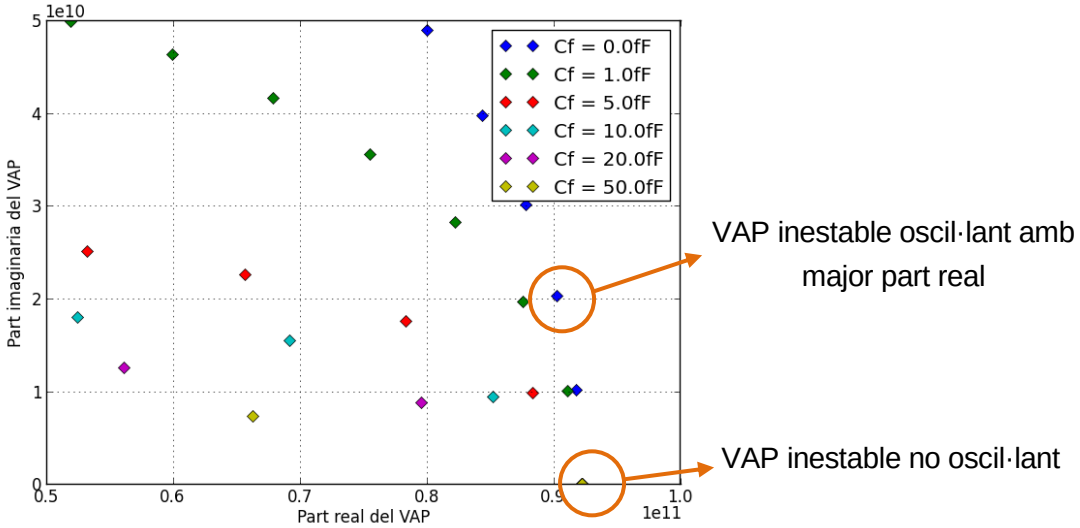


Figura 6.6: Ampliació del mapa de VAPs de varis anells biestables de 64 inversors amb filtres IOCF de diferent capacitat.

El paràmetre $\beta$ definit a l'equació 6.18 permet aproximar quan es produeix la cancel·lació d'oscil·lacions. Aquest paràmetre compara la part real del VAP no oscil·lant inestable amb el màxim de les parts reals dels VAPs oscil·lants inestables. Les figures 6.8 i 6.9 mostren el valor d'aquest paràmetre en funció de la capacitat del filtre IOCF per a un anell biestable de 64 inversors i 32 inversors. Experimentalment s'ha observat, per a anells biestables de 16, 32 i 64 inversors, que quan aquest paràmetre és inferior a $\beta < 0{,}8$ s'aconsegueix una cancel·lació completa de les oscil·lacions. S'utilitzarà aquest criteri per a estimar l'estabilitat del sistema.

$$\beta = \frac{Re(VAP_{inestable\ no\ oscil{\cdot}lant})}{\max\left(Re(VAP_{inestables\ oscil{\cdot}lants})\right)} \qquad \text{(eq 6.18)}$$

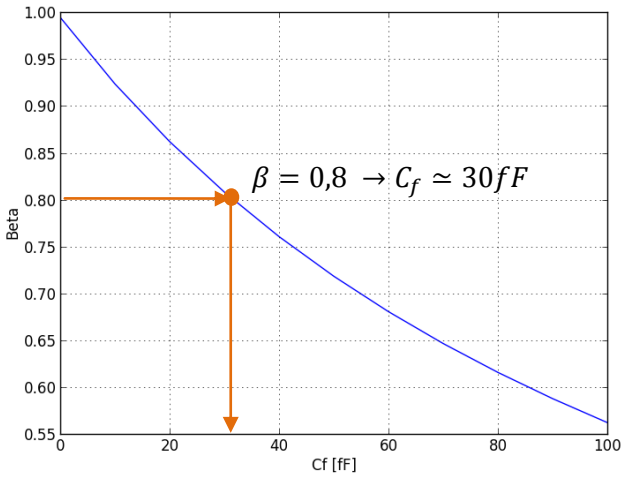


Figura 6.8: Evolució del paràmetre $\beta$ en funció de la capacitat del filtre IOCF en un anell biestable de 64 inversors

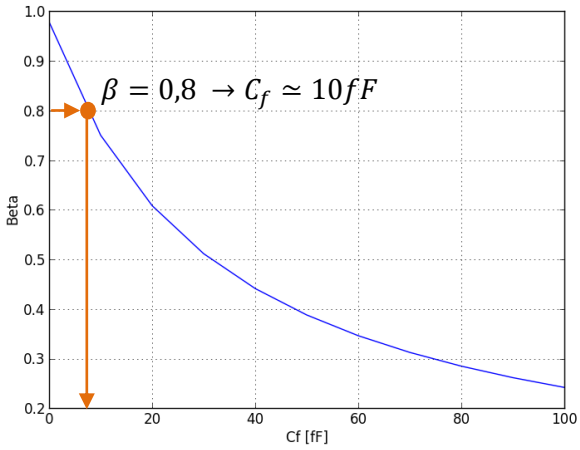


Figura 6.9: Evolució del paràmetre $\beta$ en funció de la capacitat del filtre IOCF en un anell biestable de 32 inversors

### 6.2.2. Anàlisi de la *response* del model matemàtic del BR-PUF amb IOCF

Per tal de caracteritzar el BR-PUF amb filtre IOCF (de capacitats 0 ,1,5,10,20 i 50fF) s'han simulat 50 BR-PUFs de 32 inversors als quals se'ls hi ha aplicat els mateixos 128 *challenges*. El primer que cal destacar de les simulacions és que amb el filtre s'aconsegueix cancel·lar les oscil·lacions dels anells biestables, fent que aquests arribin abans a un estat

estable. Aquest fenomen és observable en la figura 6.10, on es mostra que el 100% dels BR-PUFs amb filtre IOCF de 50fF simulats s'han estabilitzat als 20 ns.

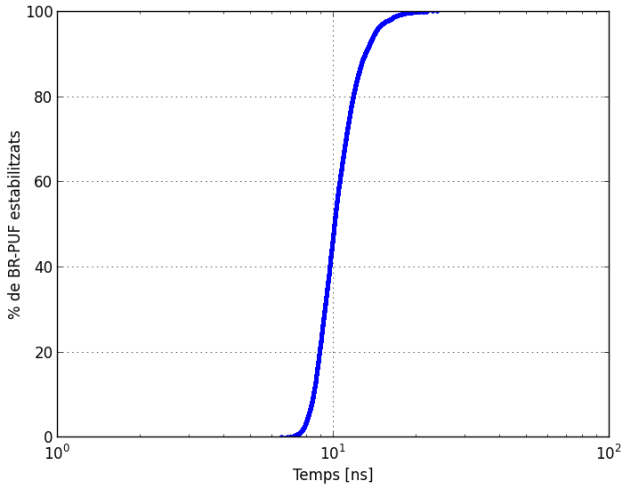


Figura 6.10: Funció de distribució acumulada del temps del temps d'estabilització dels anells d'inversors amb filtre IOCF de 50fF

Mitjançant les equacions 4.2, 4.5, i 4.6 s'analitzen les *responses* obtingudes en els BR-PUFs simulats. Cal destacar que la fiabilitat del PUF no es calcula ja que el model no inclou la possibilitat d'afegir variacions generades per pertorbacions externes. La taula 6.1 mostra els valors de les propietats analitzades per a diferents valors del filtre IOCF, on es veu que aquests no afecten de forma significativa a les propietats del PUF. Es manté una distribució aleatòria en la uniformitat dels PUFs simulats, però la resta de paràmetres tenen valors molt pròxims als òptims.

| PUF | Cf (fF) | Unicitat | Desviació estàndard *inter-distance* | Mitja Uniformitat | Desviació estàndard Uniformitat | Mitja *Bit-Aliasing* | Desviació estàndard *Bit-Aliasing* |
|---|---|---|---|---|---|---|---|
| 32 *Challenge* Bits BR - PUF | 0 | 50,25 | 0,147 | 48,02 | 26,37 | 48,02 | 5,83 |
| | 1 | 49,88 | 0,196 | 45,77 | 30,85 | 45,76 | 5,23 |
| | 5 | 49,88 | 0,157 | 55,2 | 27,07 | 55,2 | 5,37 |
| | 10 | 49,94 | 0,15 | 44,63 | 26,44 | 44,63 | 4,92 |
| | 20 | 50,19 | 0,189 | 48,69 | 30,35 | 48,69 | 6,26 |
| | 50 | 50,19 | 0,138 | 47,53 | 23,39 | 47,53 | 5,89 |

Taula 6.1: Caracterització de BR-PUFs de 32 bits de *challenge* amb filtres IOCF de diferent capacitat

# 7. TBR-PUF: *Twisted* BR-PUF

Mitjançant la utilització de l'estructura TBR-PUF descrita a l'apartat 5.3.2, es pretén solucionar la distribució aleatòria del la uniformitat del BR-PUF convertint-la en una distribució gaussiana centrada al 50%. Per a aquesta estructura, el *challenge* selecciona, en una etapa de dos inversors, quin inversor forma part de la cadena superior d'inversors i quin forma part de la cadena inferior. Un cop formades les cadenes d'inversors s'uneixen pels extrems formant un anell biestable tal i com es mostra a la 5.15.

Cal destacar que per a $N$ bits del *challenge*, el TBR-PUF necessita $2N$ inversors. A l'aplicar-li un *challenge*, es forma un anell biestable de $2N$ inversors ja que tots els inversors de l'estructura s'uneixen formant una anell biestable.

S'ha estudiat l'estructura TBR-PUF utilitzant únicament les simulacions en Python del model matemàtic del TBR-PUF. La raó de no realitzar simulacions SPICE sobre el TBR-PUF és que per a aquest cas, estudiar amb exactitud els transitoris de les oscil·lacions no aporta informació nova al funcionament dels TBR-PUFs. Aquest fet es degut a que al seleccionar un *challenge*, tant en el BR-PUF com en el TBR-PUF es forma un anell d'inversors que té un comportament independent de com s'hagi format. En canvi la caracterització del TBR-PUF permet saber si el problema de la uniformitat s'ha solucionat. Per tant obtenir un gran nombre de CRPs mitjançant simulacions del model matemàtic és el més adient per a l'estudi del TBR-PUF.

## 7.1. Estudi del TBR-PUF mitjançant el model matemàtic

El model matemàtic utilitzat per a simular el TBR-PUF és el mateix model descrit a l'apartat 6.2, però adaptant-lo al funcionament del TBR-PUF. En comptes de seleccionar un inversor d'una parella d'inversors, en el TBR-PUF, el *challenge* selecciona l'ordre en que queden enllaçats els inversors dins de l'anell.

### 7.1.1. Anàlisi de la *response* del model matemàtic del TBR-PUF

Per tal de caracteritzar el TBR-PUF s'han simulat 50 TBR-PUFs de 64 inversors (tenen 32 bits de *challenge*) als quals se'ls hi ha aplicat els mateixos 128 *challenges*. El primer que cal destacar és que, com es mostra en la figura 7.1, el problema que tenia el BR-PUF amb la uniformitat desapareix, segueix una distribució gaussiana amb una mitja del 50,47%. Aquesta distribució s'aproxima a la distribució òptima per a la uniformitat. Es confirma doncs que aquesta arquitectura soluciona el problema de la uniformitat.

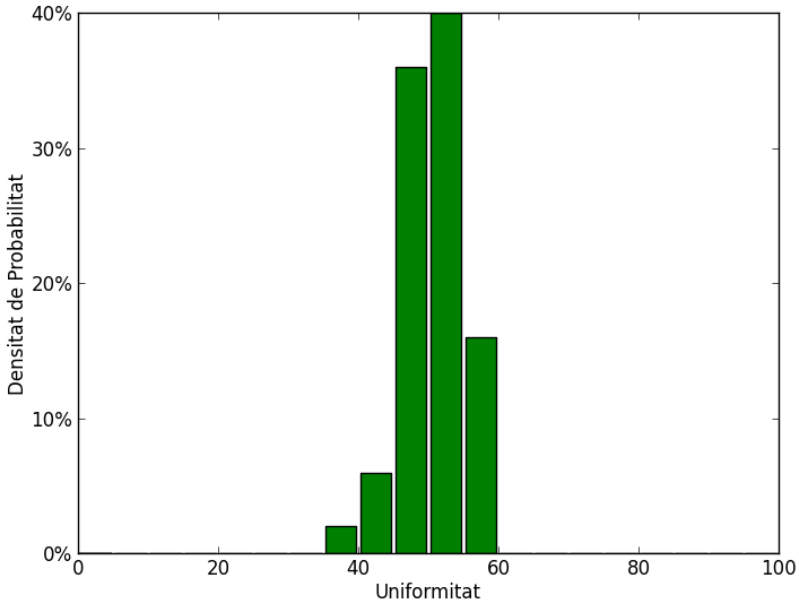


Figura 7.1: Histograma de les uniformitats dels TBR-PUFs simulats amb 32 bits de *challenge*

Si es combinen l'estructura TBR-PUF amb el filtre IOCF es poden solucionar els dos grans problemes del BR-PUF. Per tal de descartar que no hi ha cap interacció negativa entre aquestes estructures, s'han simulat 50 TBR-PUFs amb filtres IOCF de 32 bits de *challenge* als quals se'ls hi han aplicat els mateixos 128 *challenges*. La taula 7.1 mostra la caracterització dels TBR-PUFs amb filtres IOCF. Es es confirma doncs que el filtre IOCF no afecta a les propietats del TBR-PUF,

L'únic fet a tenir en compte és que, per al mateix nombre de bits de *challenge*, el TBR-PUF forma un anell biestable amb el doble nombre d'inversors que l'estructura BR-PUF. En consequencia, el filtre IOCF haurà de tenir una capacitat major que en el cas d'un BR-PUF.

| PUF | Cf (fF) | Unicitat | Desviació estàndard *inter-distance* | Mitja Uniformitat | Desviació estàndard Uniformitat | Mitja *Bit-Aliasing* | Desviació estàndard *Bit-Aliasing* |
|---|---|---|---|---|---|---|---|
| 32 *Challenge* Bits TBR - PUF | 0 | 50,11 | 0,086 | 50,47 | 3,96 | 50,47 | 6,64 |
| | 1 | 50,11 | 0,072 | 50,31 | 4,7 | 50,31 | 6,68 |
| | 5 | 50,08 | 0,07 | 51,11 | 5,62 | 51,11 | 6,7 |
| | 10 | 50,11 | 0,071 | 49,31 | 4,45 | 49,31 | 6,64 |

Taula 7.1: Caracterització de TBR-PUFs de 32 bits de *challenge* amb filtres IOCF de diferent capacitat

# 8. Disseny del BR-PUF per a una implementació en un prototip ASIC

En aquest capítol es descriu la implementació física del BR-PUF amb filtre IOCF que s'ha realitzat sobre un disseny *full-custom* per a ser implementat en un prototip ASIC amb tecnologia cmos065. S'ha decidit no implementar la estructura TBR-PUF per dos motius, el primer és que en aquests moments un grup d'investigació alemany està desenvolupant i aquesta arquitectura, el segon és que també interessa veure quin és l'efecte del filtre IOCF sobre la *response* del BR-PUF. S'ha dissenyat el *layout* de 6 BR-PUF amb filtres IOCF de diferents capacitats amb l'objectiu de poder caracteritzar completament el BR-PUF i poder obtenir dades experimentals. El control del ASIC es fa des d'un autòmat de control. dissenyat en una FPGA i descrit en llenguatge VHDL. La figura 8.1 mostra de forma simbòlica la interacció d'aquest dos elements.

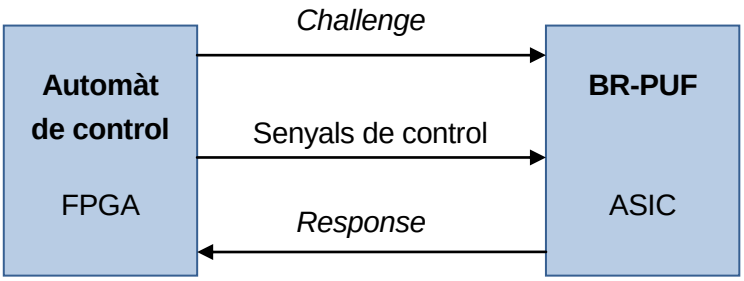


Figura 8.1: Interacció entre el BR-PUF i l'autòmat de control

S'han implementat BR-PUFs de 64 inversors amb un nombre de CRPs elevat, exactament $2^{64} = 1{,}8 \cdot 10^{19}$, que faria inviable la caracterització completa del PUF per part d'un hacker. Donat que es necessiten 64 bits pel *challenge* i que el nombre de pins del ASIC és reduït, s'ha incorporat una cadena de scan a través de la qual l'autòmat de control envia el *challenge* seriadament.

El mateix succeeix a l'hora d'extreure la *response* del PUF. El disseny d'un circuit *full-custom* que realitzi els còmput de la *response* (mitjançant la equació 5.12) seria molt laboriós. Per aquest motiu s'ha decidit enregistrar l'estat dels inversors en un cert instant de temps posterior a l'aplicació del *challenge*. Un cop enregistrats els estats es transmeten de forma seriada cap a la FPGA pel pin de "scan-out" per a processar posteriorment la cadena de bits.

Els senyals de control s'encarregan d'inicialitzar el circuit, gestionar la transmissió, l'aplicació del *challenge* i de la *response,* i seleccionar quins PUFs estan en funcionament en un

moment determinat. A més, també s'incorpora al PUF un detector d'estabilitat per tal de registrar l'instant enque s'han estabilitzat els anells d'inversors.

## 8.1. Arquitectura implementada del BR-PUF amb IOCF

En la pràctica el BR-PUF no s'implementa tal i com es mostra a la figura 5.2 ja que per obtenir cada *response* caldria desactivar i activar de nou l'alimentació dels inversors,el que faria el temps de resposta molt lenta. Altrament, els anells es contrueixen amb portes NOR que permeten inicialitzar el circuit a un estat inestable a partir d'una entrada. La figura 8.2 mostra aquesta arquitectura. Si el senyal de control "Reset_PUF" té el nivell de tensió alt ('1' lògic), llavors la sortida de les portes NOR estaria estabilitzada al nivell baix de tensió ('0' lògic). En canvi si aquesta senyal val '0' les portes NOR passen a comportar-se com a inversors, creant un anell biestable que es troba en un estat inestable (la sortida de totes les portes NOR val '0'). L'anell biestable tendirà cap a un dels dos possibles estats estables de la mateixa forma que succeiria amb un anell d'inversors, l'arquitectura incorpora també el filtre IOCF.

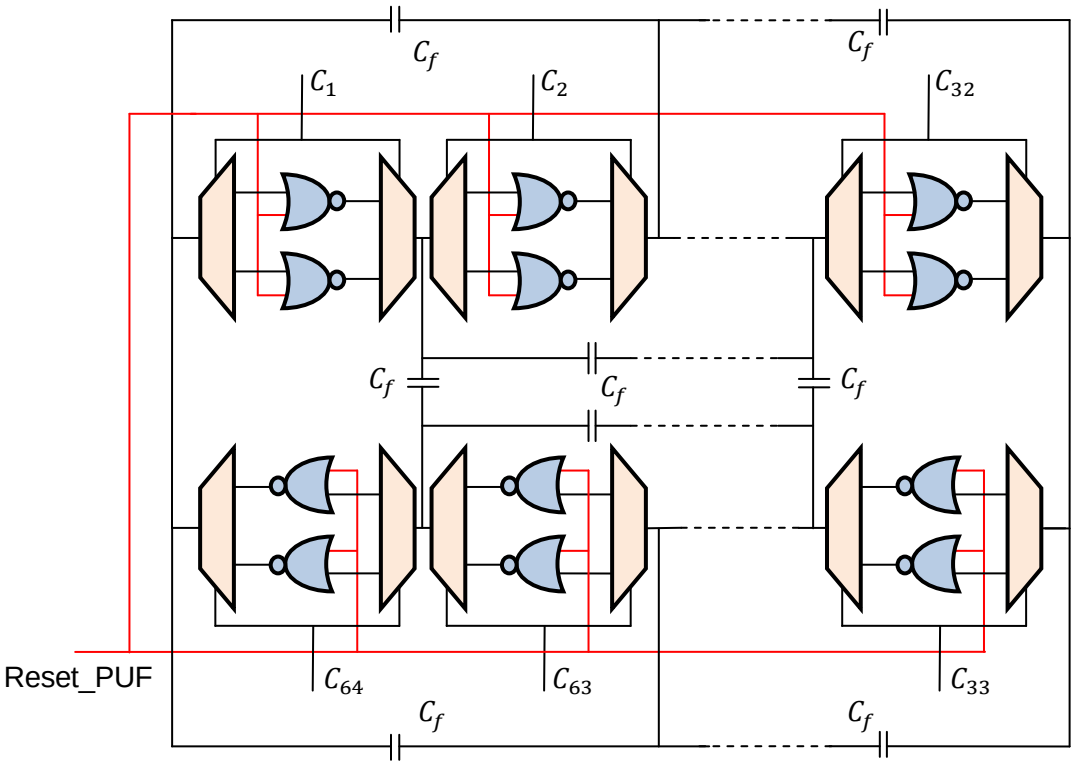


Figura 8.2: Esquema del BR-PUF amb filtre IOCF realitzat amb portes NOR per a permetre inicialitzar el circuit a un estat inestable

En el disseny del BR-PUF també s'han implementat una sèrie de registres que permeten enregistrar tant el *challenge* a aplicar com els estats dels l'inversors, el disseny també inclou els detectors d'estabilitat. En els propers apartats s'explica la implementació d'aquests sistemes i com es realitza el seu control.

## 8.2. Dimensionament del circuit

En aquest apartat s'explica com s'han dimensionat els elements crítics del BR-PUF. Primerament es parla del dimensionament de la porta lògica NOR que s'ha utilitzat en la formació de l'anell biestable. A continuació es descriu com s'ha dimensionat el filtre IOCF. Finalment es descriu el disseny i el dimensionament del detector d'estabilitat. La resta de components que formen el disseny del BR-PUF s'han dimensionat d'acord amb les regles estàndard de la tecnologia cmos065.

### 8.2.1. Dimensionament de les portes NOR

La porta lògica NOR és el component principal del BR-PUF, la figura 8.3 mostra el seu esquema. El terminal d'entrada A correspon a la entrada del senyal de control "Reset_PUF", els transistors corresponents a aquest terminal s'han dimensionat amb les mateixes mides que les d'una cel·la estàndard. En canvi, als transistors corresponents al terminal d'entrada B se'ls ha augmentat la longitud de canal fins a 7 vegades la mínima de la tecnologia (0,42µm) a fi de limitar la freqüència d'oscil·lació de l'anell biestable.

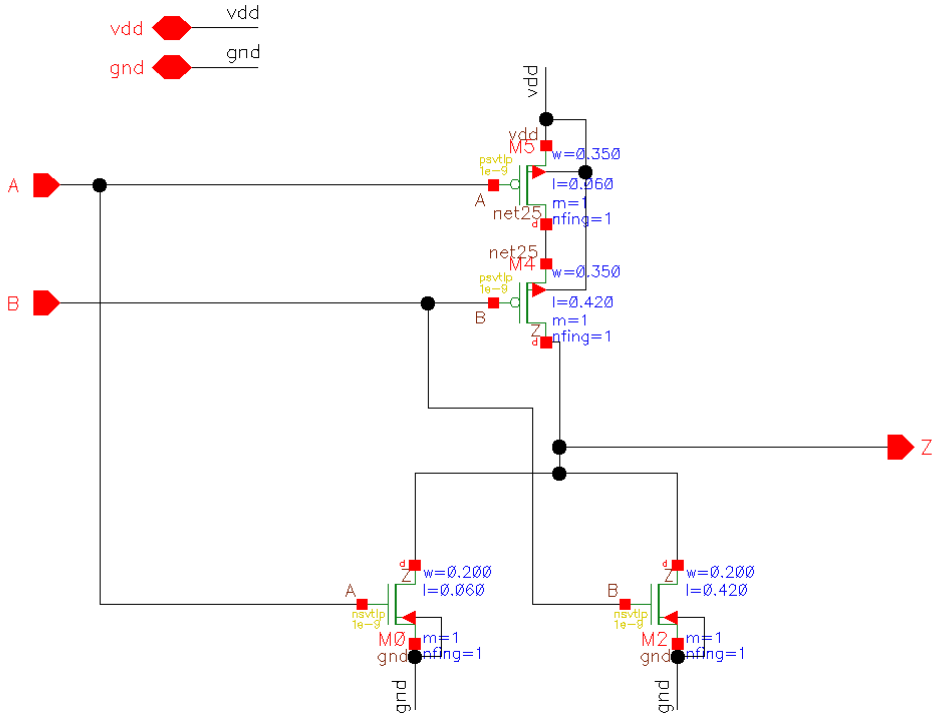


Figura 8.3: Esquemàtic de la porta NOR

### 8.2.2. Dimensionat dels condensadors del filtre IOCF

El dimensionat del filtre IOCF s'ha realitzat seguint els següents passos:

#### 8.2.2.1. Parametrització de la porta NOR

Suposant que s'aplica un nivell de tensió baix a la entrada corresponent a la senyal de *Reset_PUF*, s'ha parametritzat el comportament de la porta seguint els passos descrits a l'apartat 5.2.1.1. S'han obtingut els següents valors $g_m = 4{,}21 \cdot 10^{-4}$, $g_d = 4{,}84 \cdot 10^{-5}$i $C = 10^{-15}$.

#### 8.2.2.2. Estudi dels VAPS del anell biestable

A partir del model matemàtic descrit a l'apartat 6.2.1 s'ha trobat la corba del paràmetre $\beta$ (equació 6.18) en funció de la capacitat del filtre IOCF. La figura 8.4 mostra aquesta corba, i es pot observar que per a una capacitat del condensador de 70fF el paràmetre $\beta$ es inferior a 0,8. A continuació s'ha comprovat amb Spice que per a aquest valor del paràmetre $\beta$, les oscil·lacions es cancel·len pràcticament en la seva totalitat.

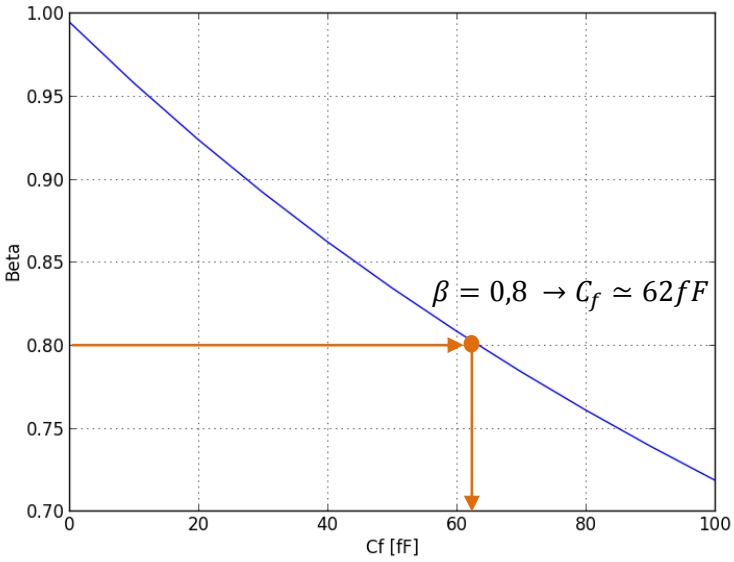


Figura 8.4: Variació del paràmetre $\beta$ en funció de la capacitat del filtre IOCF

#### 8.2.2.3. Selecció de la capacitat del condensador

Tenint en compte que el condensador que s'utilitzarà en la creació del filtre té una capacitat mínima de 44fF, s'ha decidit implementar 6 BR-PUFs amb filtres de les següents capacitats: 0fF, 44fF, 50fF, 60fF, 70fF 80fF. D'aquesta manera es podran estudiar els seus efectes en la *response* del BR-PUF i també es podrà observar la disminució del temps d'estabilització.

#### 8.2.2.4. Comprovació de la cancel·lació d'oscil·lacions

La cancel·lació de les oscil·lacions s'ha observat realitzant 200 simulacions Monte Carlo d'un anell biestable format per 64 portes NOR amb un filtre IOCF de $C_f = 80fF$ , la figura 8.5 mostra l'evolució dels transitoris. Es pot observar que en cap d'elles hi apareixen oscil·lacions.

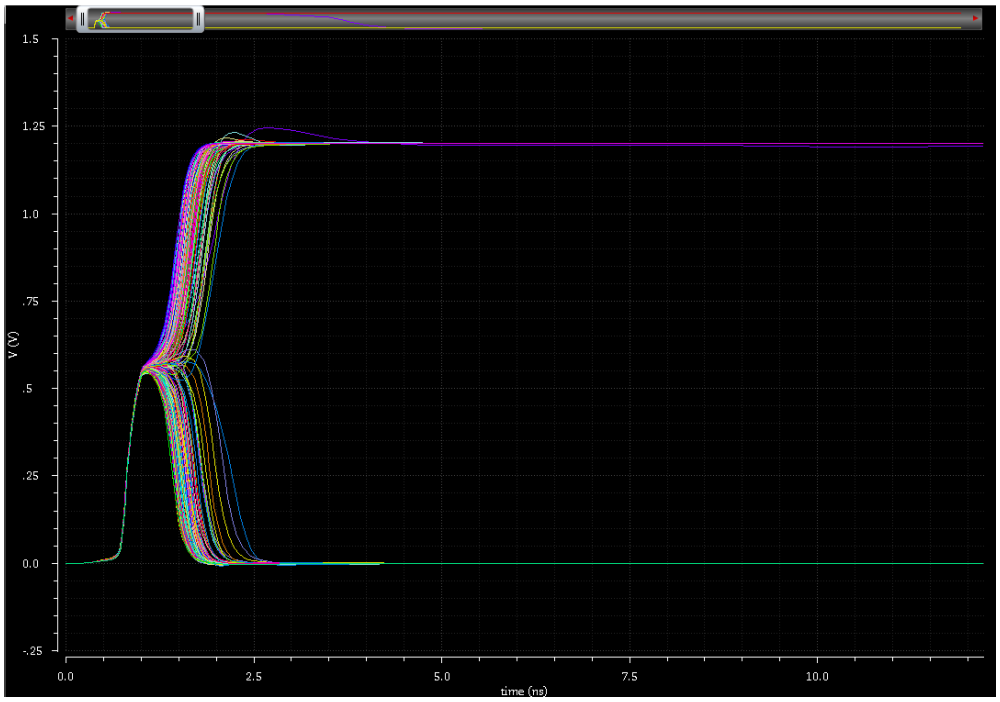


Figura 8.5: Evolució de la sortida d'una porta NOR en la simulació de 200 anells biestables amb filtre IOCF de $C_f = 80fF$

### 8.2.3. Dimensionament del detector d'estabilitat

El detector d'estabilitat té la funció de determinar l'instant en el qual l'anell biestable s'ha estabilitzat. Aquesta funció utilitzat un circuit amb una arquitectura similar a la de la figura 8.6. Quan a la entrada del detector hi ha un nivell de tensió alt, el condensador es carrega ja que els seus terminals es troben entre la tensió d'alimentació i el terra. En canvi, a l'aplicar un nivell de tensió baix el condensador es descarrega a través de la resistència. Quan el senyal d'entrada del detector és oscil·lant, el condensador es descarrega i carrega periòdicament. Dimensionant el circuit de forma adequada, es pot aconseguir que el procés de carrega sigui més ràpid que el de descarrega i que a la sortida hi apareix un nivell de tensió baix mentre l'entrada del detector oscil·li,. En canvi quan el senyal s'estabilitzi a un nivell de tensió baix, la sortida de l'inversor es generarà un nivell de tensió alt.

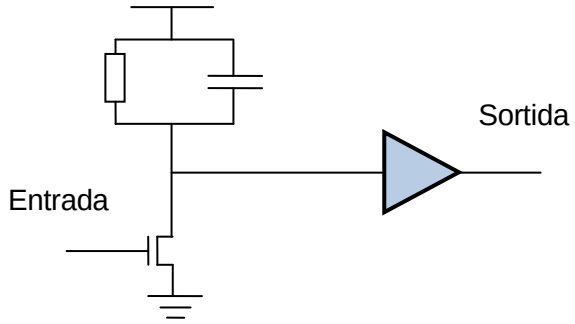


Figura 8.6: Estructura funcional del detector d'estabilitat

A l'hora d'implementar aquest sistema en un circuit electrònic, es pot prescindir de la resistència i el condensador si s'empra un transistor p en saturació i es dimensiona correctament el *buffer* de sortida. El transistor en saturació es comporta com la resistència i les capacitats paràsites dels transistors i del *buffer* actuen com a condensador. La figura 8.7 mostra l'esquemàtic final del detector d'estabilitat.

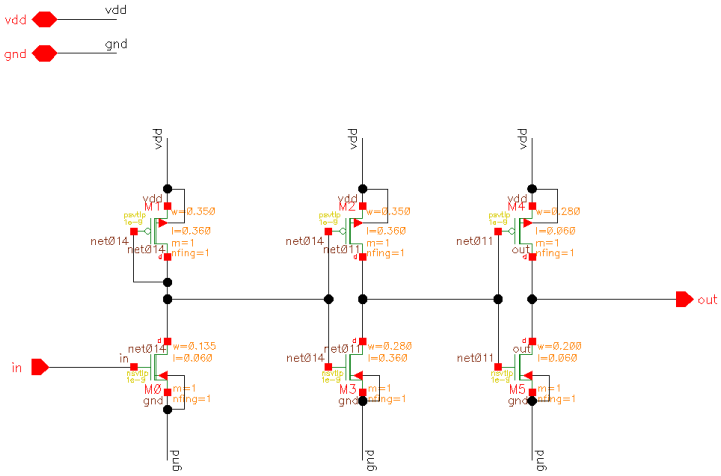


Figura 8.7: Esquema de transistors del detector d'estabilitat

S'han realitzat simulacions SPICE del detector d'estabilitat quan a la seva entrada se li aplica un senyal oscil·lant a 5GHz, que és aproximadament la freqüència d'oscil·lació d'un anell biestable sense filtre IOCF. La figura 8.8 mostra l'evolució d'aquestes simulacions. Mentre la senyal d'entrada oscil·la (senyal vermell), la capacitat parasitària (senyal groc) es carrega i descarrega contínuament. La tensió de sortida del detector (senyal verd) es manté en un nivell baix fins que les oscil·lacions s'aturen, moment en el qual passa a tenir un nivell de tensió alt.

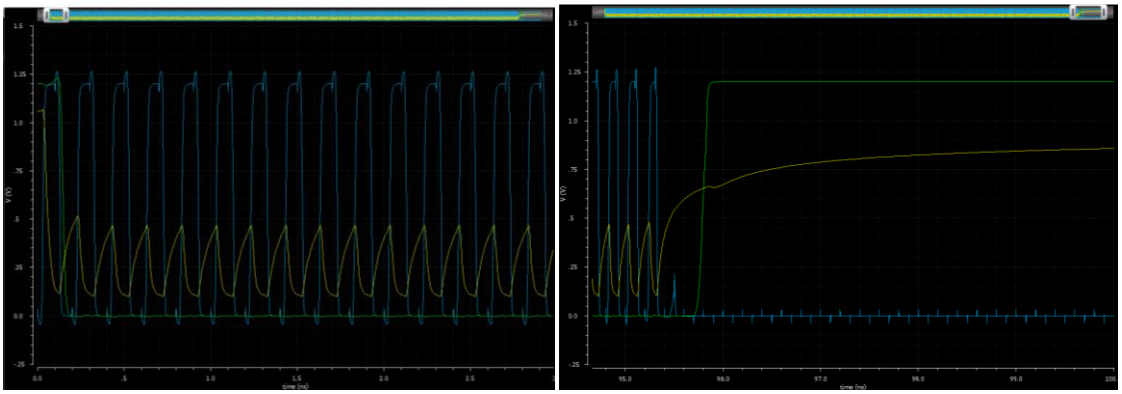

Figura 8.8: Simulació del detector d'estabilitat. El senyal vermell correspon a l'entrada del detector; el senyal groc correspon a la tensió d'entrada al *buffer* del detector i el senyal verd correspon a la sortida del detector

Cal notar que aquest sistema nomes detecta les oscil·lacions que s'estabilitzen en un nivell de tensió baix. Per ampliar-ho també a nivell alt s'ha afegit un detector a la sortida de dos etapes del anell biestable consecutives tal i com es mostra a la figura 8.9. El detector d'estabilitat es completa utilitzant una porta OR que està connectada als dos detectors. A més, s'utilitzen dos detectors d'estabilitat consecutius, en comptes d'un, per evitar que es detecti la estabilitat de forma errònia degut al soroll en les oscil·lacions.

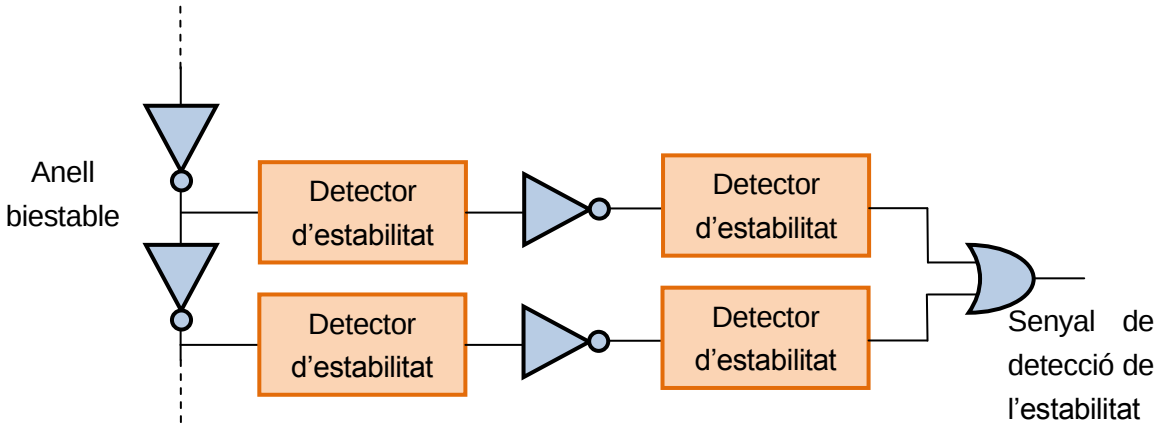


Figura 8.9: Implementació del detector d'estabilitat complert en el BR-PUF

El detector presenta un retard d'aproximadament de 1 ns edesde que l'anell s'estabilitza i que el senyal de sortida del detector canvia de valor.

## 8.3. Disseny del prototip

S'han implementat els 6 BR-PUFs dividint el sistema complert en elements unitaris, anomenats "BR_PUF_SLICE". Cada "BR_PUF_SLICE" conté dues etapes (anomenades

"BR_PUF_STAGE") per a cada un dels 6 BR-PUF a implementar. La figura 8.10 mostra la divisió del disseny del BR-PUF en elements iterables. Mitjançant la unió de varis d'ells es creen els anells biestables del BR-PUF.

Cada cel·la "BR_PUF_STAGE" conté dues etapes (dues portes NOR) del anell biestable com s'explicarà més endavant. Per tant, amb la unió de 16 cel·les "BR_PUF_SLICE" s'obtenen anells biestables de 64 etapes.

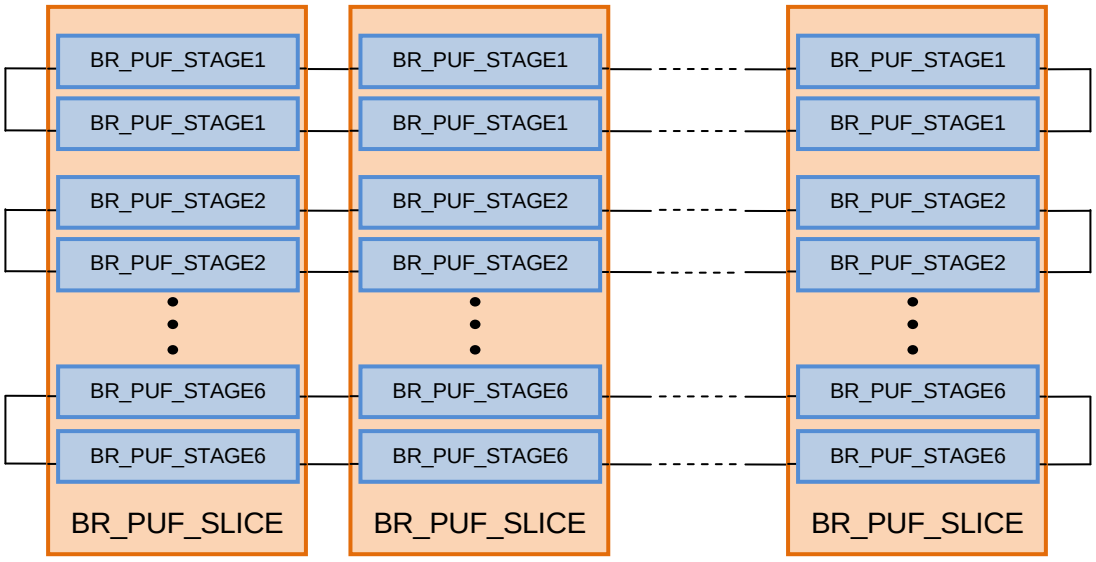


Figura 8.10: Divisió del disseny del BR-PUF en elements iterables

En els pròxims apartats es descriu com s'han dissenyat els elements iterables i els senyals de control necessàris pel funcionament d'aquests.

### 8.3.1. Disseny d'una etapa del BR-PUF

Cada cel·la "BR_PUF_STAGE" conté dues etapes del anell biestable, amb els corresponents condensadors del filtre IOCF, el detector d'estabilitat descrit a l'apartat 8.2.3 i dos registres *latch* que emmagatzemen l'estat dels inversors. La figura 8.11 mostra el seu disseny. Cal destacar que s'han creat 6 cel·les diferents amb condensadors de capacitats de 0fF, 44fF, 50fF, 60fF,70fF i 80fF.

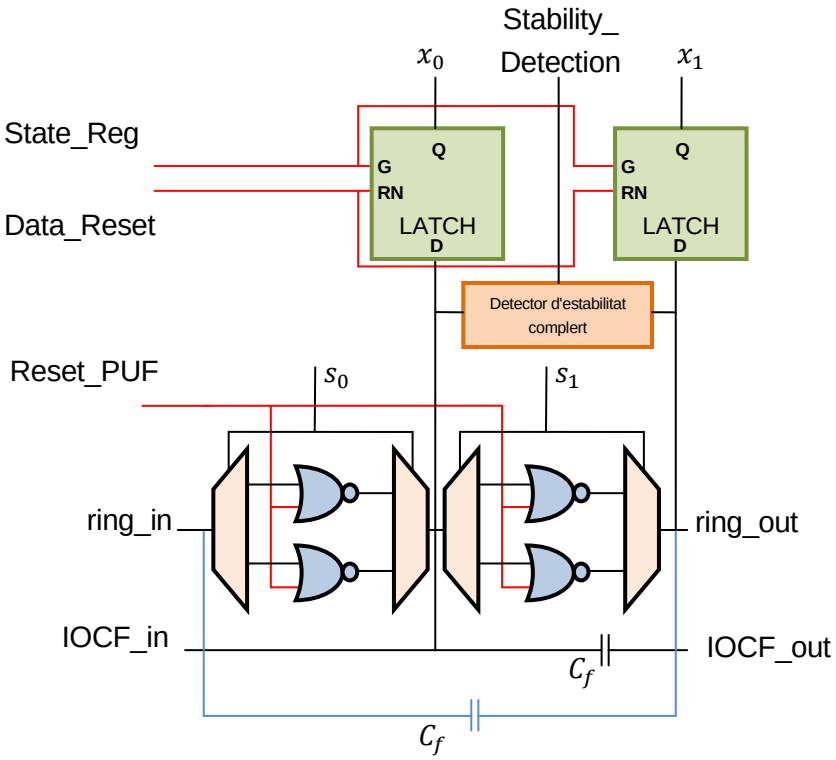


Figura 8.11: Disseny de la cel·la "BR_PUF_STAGE"

Els terminals "ring_in", "ring_out", "IOCF_in" i "IOCF_out" permeten la iteració d'aquesta cel·la. Unint varies instàncies d'aquesta encadenant els ports "ring_in" amb els "ring_out" i els ports "IOCF_in" amb els "IOCF_out" es forma l'anell biestable amb filtre IOCF. Els terminals de sortida $x_0$ i $x_1$ transmeten l'estat dels inversors enregistrat pels registres tipus *latch*.

Les senyals de control $s_0$ i $s_1$corresponen als bits de *challenge*. En funció del seu valor es selecciona quina porta NOR forma part de l'anell biestable. La senyal "Reset_PUF" permet activar o desactivar l'anell biestable, inicialitzant-lo sempre a un estat inestable. "State_Reg" és el senyal encarregat d'indicar al *latch* que s'ha d'enregistrar l'estat dels inversors. La senyal "Data_Reset" realitza un *reset* asíncron actiu per nivell baix als registres.

### 8.3.2. Disseny de l'element unitari del sistema

L'element unitari que s'itera per a formar els 6 BR-PUFs és la cel·la "BR_PUF_SLICE". Com ja s'ha explicat aquesta cel·la està formada per 12 cel·les "BR_PUF_STAGE" (dues per a cada BR-PUF a implementar). A part de les cel·les "BR_PUF_STAGE", aquest element necessita 2 sistemes addicionals per al seu funcionament. Per una banda és necessari poder extreure els estats dels inversors enregistrats a les cel·les "BR-PUF_Stage". A més

també ha de ser possible la introducció del *challenge* que s'aplica a totes les etapes del BR-PUF.

Els estats dels inversors s'extrauen mitjançant un sistema que transmet aquesta informació de forma seriada a través d'una cadena d'scan, veure-ho a la figura 8.12. Un multiplexor selecciona el mode captura o desplaçament de la cadena. Si el senyal de control "Scan_Out_Select" val '0', el registre està en el mode captura mentre que per '1' es troba en el mode desplaçament. Els bits a extreure mitjançant la cadena de scan es transmeten a través dels terminals de "Scan_Out_Input" i de "Scan_Out_Output" de dos cel·les "BR_PUF_SLICE" consecutives. El senyal "Data_Reset " permet realitzar un *reset* als registres del sistema. Els bits d'informació es transmeten cada cop que es produeix un flanc positiu en la senyal "Scan_Out_Clk".

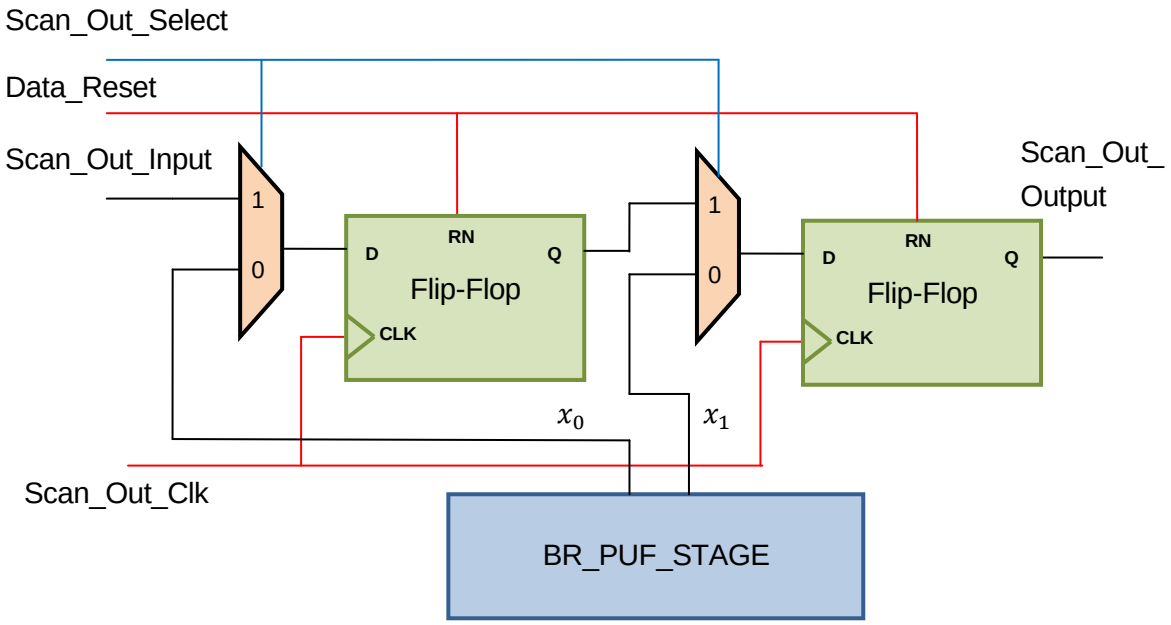


Figura 8.12: Cadena d'scan de cada cel·la "BR_PUF_STAGE"

Una cadena d'scan permet la introducció del *challenge* de forma seriada tal i com es mostra a la figura 8.13. La cadena s'activa pel flanc positiu del senyal "Scan_In_Clk. Unint el terminal de "Scan_In_Output" d'una cel·la "BR_PUF_SLICE" amb el terminal "Scan_In_Input" s'exten la cadena a totes les cel·les, permetent que el *challenge* es pugui introduir de forma seriada. Un registre latch intern manté els valors d'entrada mentre es fan lliscar els bits per la cadena. Aquests actualitzen el seu valor quan el senyal "Data_Reg" té un nivell alt de tensió. La senyal de "Data_Reset", activada per nivell baix, realitza un *reset* a tots els registres.

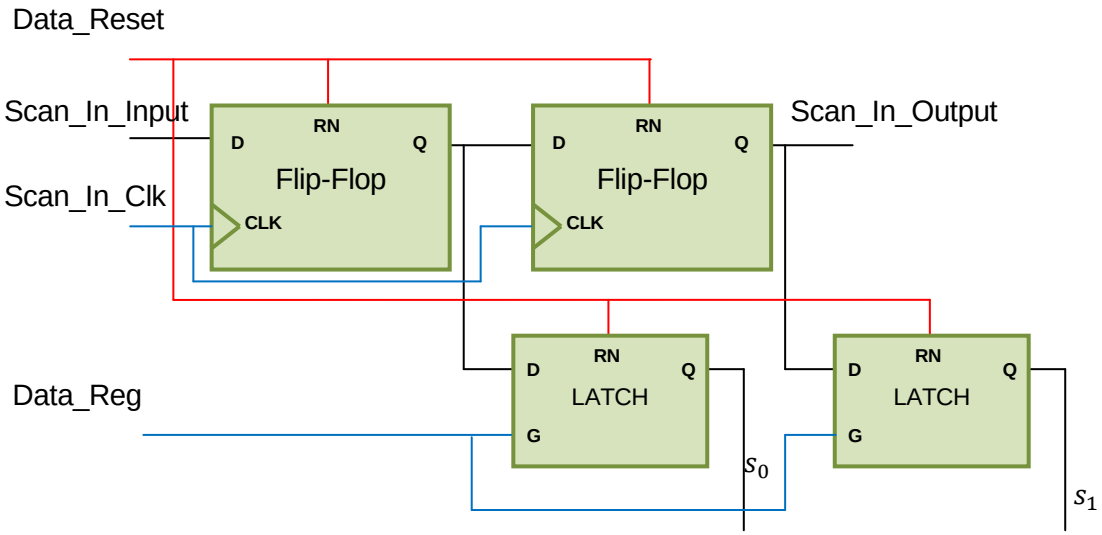


Figura 8.13: Cadena d'scan encarregada d'introduir el *challenge* al BR-PUF

Totes les senyals de control que arriben al "BR_PUF_SLICE" actuen sobre un nombre de portes molt elevat. Per aquest motiu s'utilitzen *buffers* amb un *fan-out* suficientment gran perquè les senyals puguin actuar sobre totes les portes.

### 8.3.3. Disseny del xip

La cel·la que conté el sistema complet s'anomena "BR_PUF". Aquesta cel·la està formada per 16 cel·les "BR_PUF_SLICE" i un sistema encarregat de seleccionar els PUFs que estaran en funcionament en un moment determinat. D'aquesta forma s'han construït 6 BR-PUFs de 64 inversors.

Les cel·les "BR_PUF_SLICE" s'han unit com s'ha explicat en l'apartat anterior per poder formar els anells biestables i les cadenes de transmissió de dades (tant d'entrada com de sortida). A més s'han utilitzat *buffers* per permetre que els senyals de control tinguin prou força com per actuar sobre les cel·les "BR_PUF_SLICE".

L'activació dels PUFs que es troben en funcionament es fa a través de la mateixa cadena d'scan que envia el *challenge*. El sistema es molt similar al de la figura 8.13; la cadena d'scan s'activa pel flanc positiu de la senyal "Scan_In_Clk". Per separar la cadena de dades de la resta del circuit, s'han utilitzat registres tipus *latch* que emmagatzemen la informació dels registres *flip-flop* quan la senyal "Data_Reg" té un nivell de tensió alt. Quan s'enregistra un '1' en un cert registre *latch*, i la senyal "Run_PUF" val '1', la senyal "Reset_PUF" passa de valer '0' a valer '1'. D'aquesta manera, el BR-PUF corresponent al registre es posa en funcionament. La senyal de "Data_Reset" permet realitzar un *reset* asíncron als registres. El sistema està format per 6 etapes de *flip-flop*, *latch* i porta NAND.

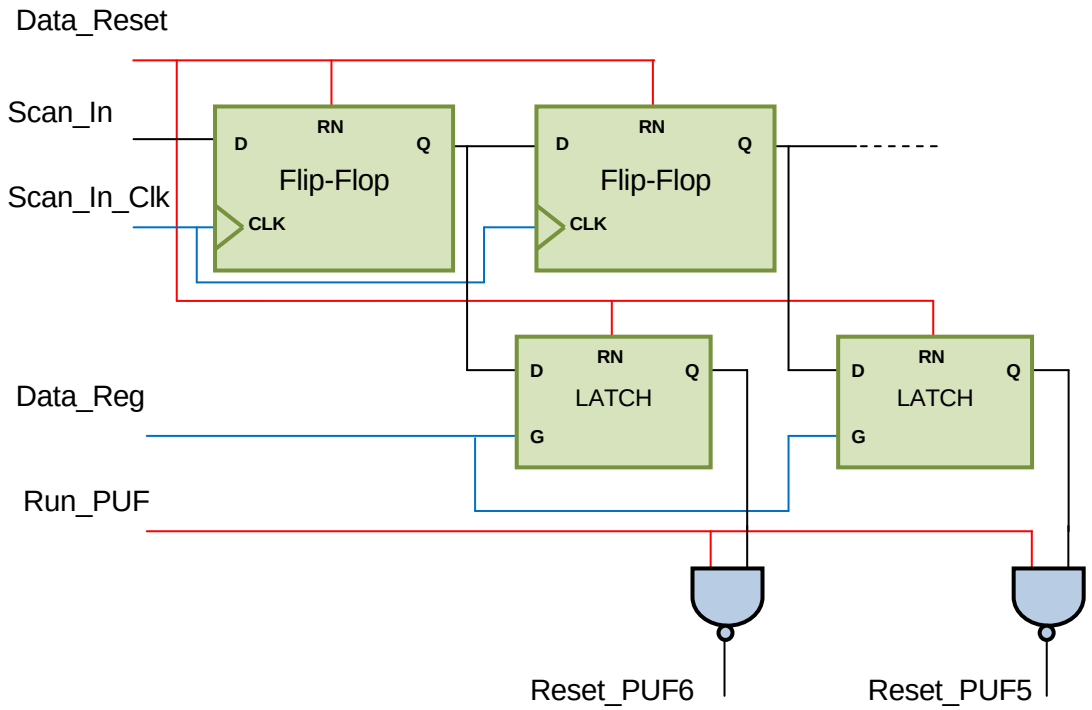


Figura 8.14: Sistema parcial encarregat d'habilitar el funcionament dels PUFs. El sistema complet està format per 6 etapes

Per poder comunicar el circuit ASIC amb els pins de l'encapsulat, s'utilitzen pads d'entrada i sortida de la llibreria estàndard IO65LPHVT_SF_1V8_50A_7M4X0Y2Z. La cel·la "BR_PUF_pad" conté la cel·la "BR_PUF" i els pads corresponents als senyals d'entrada i sortida. S'han emprat 19 pads: 7 corresponen als senyals de control ("Scan_Out_Clk", "Scan_In_Clk", "Run_PUF", "State_Reg", "Scan_Out_Select", "Data_Reset"i "Data_Reg"); 1 correspon al terminal ("Scan_in") que permet la introducció, de forma seriada, del *challenge* i dels PUFs en funcionament; 6 corresponen a les senyals de detecció d'estabilitat dels 6 BR_PUFs ("Stability_Detection"); 1 és el terminal per el qual s'extreuen ("Scan_Out"), de forma seriada, els estats dels inversors; 2 són l'alimentació del circuit ("terra circuit" i "alimentació circuit"); i 2 són l'alimentació dels pads ("terra pads" i "alimentació pads"). La distribució dels pads es mostra en la figura 8.15.

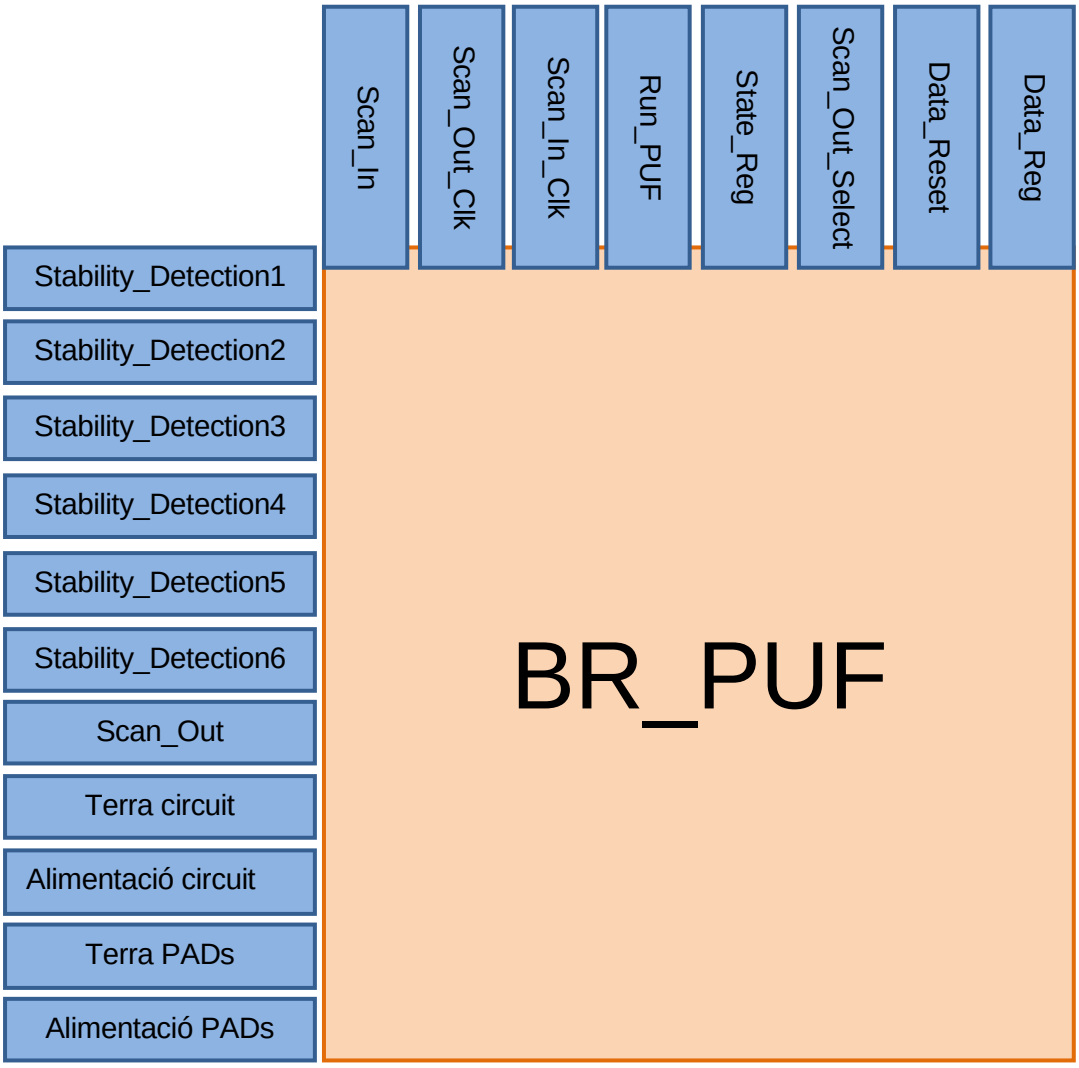


Figura 8.15: Distribució dels pads al voltant de la cel·la "BR_PUF"

El nucli del circuit va alimentat a 1,2V mentre que els pads van alimentats a una tensió de 1,8V. El disseny del circuit complet ocupa una àrea de $0{,}512mm\ x\ 0{,}6726mm = 0{,}344mm^2$). La figura 8.16 mostra el *layout* del ASIC corresponent al sistema “BR_PUF” amb els pads. Els dissenys *layouts* de la resta de cel·les es troba a l'annex E. Els esquemàtics d'aquestes cel·les es troben a l'annex D.

Figura 8.16: *Layout* del ASIC corresponent al sistema "BR_PUF" amb els pads d'entrada i sortida

## 8.4. Interfície de control del circuit

Com ja s'ha explicat, la part de control del circuit s'ha realitzat de forma externa mitjançant un autòmat de control síncron implementat en una FPGA. Aquesta FPGA s'encarrega d'introduir el *challenge* al prototip i de generar les senyals de control. La FPGA és la Cyclone III de la placa de desenvolupament DE0 de Terasic. En aquest apartat es farà un resum dels senyals de control del circuit i es descriurà la implementació de l'autòmat.

### 8.4.1. Senyals d'entrada i sortida del BR-PUF

En aquest apartar es mostra un resum dels senyals d'entrada i sortida necessàris pel funcionament del circuit. Cal destacar que els senyals dels detectors d'estabilitat no s'han emprat en el control del circuit ja que són únicament d'observació del temps d'estabilització i s'empraran per a poder graficar les funcions de distribució acumulades dels temps d'estabilització des anells biestables.

#### 8.4.1.1. "Scan_In"

Senyal formada per una cadena de bits, encarregada de seleccionar el *challenge* a aplicar als 6 BR-PUFs i de seleccionar quins dels PUFs funcionen en un determinat moment. L'entrada dels bits del *challenge* i dels bits de selecció del PUF en funcionament es realitza de forma seriada. La llargada total de la cadena de bits és de 70 bits. Els primers 64 bits introduïts corresponen al *challenge*. Els següents 6 bits corresponen a la selecció del PUF a utilitzar. El circuit s'ha dissenyat de forma que els bits menys significatiu, del grup de 6 bits de selecció de PUF, corresponguin al primer BR-PUF implementat (sense filtre IOCF). El bit més significatiu correspon al BR-PUF que té un filtre IOCF de 80 fF.

#### 8.4.1.2. "Scan_In_Clk"

Senyal de rellotge que transmet la cadena de bits corresponent a l'entrada de dades d'un *flip-flop* al següent. Aquesta senyal actua sobre tots els registres tipus *flip-flop* corresponents a la cadena de bits d'entrada al circuit. Quan es produeix un flanc positiu en aquest senyal la cadena es desplaça 1 bit..

#### 8.4.1.3. "Data_Reg"

Senyal encarregat d'enregistrar els bits de la cadena de dades d'entrada. Quan aquest senyal té un nivell de alt s'actualitzen els registres tipus *latch* amb el contingut dels *flip-flop* de la cadena corresponents.

#### 8.4.1.4. "Run_PUF"

Senyal encarregat de posar en funcionament els anells biestables. Al posar-la a nivell alt, els senyals interns "Reset_PUF" corresponents als PUFs seleccionats per a funcionar passen del valor '1' al valor '0'. D'aquesta manera les portes NOR que del BR-PUF passen a comportar-se com a inversors formant l'anell biestable.

#### 8.4.1.5. "State_Reg"

Senyal encarregat d'enregistrar els bits de la cadena de dades d'entrada. Quan aquesta senyal té un nivell de tensió alt, s'enregistra la sortida de tots els inversors del circuit en registres tipus *latch*.

#### 8.4.1.6. "Scan_Out_Select"

Senyal encarregat de seleccionar quina informació es volca als registres *flip-flop* corresponents a la sortida de dades del circuit. Si té un nivell de tensió baix, la informació que arriba als registres *flip-flop*, és l'estat del inversor enregistrat pels registres *latch* de la cel·la "BR_PUF_STAGE". En canvi si aquest senyal té un nivell de tensió baix, els bits d'informació es transmeten de *flip-flop* en *flip-flop*.

#### 8.4.1.7. "Data_Reset"

Senyal de *reset* asíncron dels registres del BR-PUF. Aquesta senyal s'activa per nivell baix.

#### 8.4.1.8. "Scan_Out_Clk"

Senyal de rellotge que transmet la cadena de bits corresponent a la sortida de dades del d'un *flip-flop* al següent. Aquesta senyal actua sobre tots els registres tipus *flip-flop* corresponents a la cadena de bits que conté la informació dels estats dels inversors. Quan es produeix un flanc positiu en aquesta senyal es transmet el bit d'informació d'un registre al següent.

#### 8.4.1.9. "Scan_Out"

Senyal de sortida del circuit que transmet de forma seriada els estats dels inversors. La informació de sortida és una cadena de 384 bits (64 bits per cada un dels 6 BR-PUFs). Es transmeten les dades en grups de 64 bits corresponents als diferents BR-PUFs implementats, l'ordre de transmissió és descendent. Els 64 primers bits corresponen als estats dels inversors del 6è BR-PUF (amb filtre IOCF de 80 fF). Els últims 64 bits corresponen als estats dels inversors del 1er BR-PUF (sense filtre IOCF).

#### 8.4.1.10. “Stability_Detection”

Senyal encarregat de senyalitzar l’instant en el que s’ha estabilitzat un anell biestable. N’hi ha una per a cada BR-PUF. Quan aquest senyal es troba en nivell baix indica que l’inversor està oscil·lant, mentre que quan es troba a nivell alt de tensió indica que l’anell biestable s’ha estabilitzat.

### 8.4.2. Autòmat de control

L'autòmat de control permet gestionar tots els senyals de control a fi de: introduir el *challenge*, posar en funcionament dels anells biestables i extreure la *response*. Per a realitzar aquesta funció s'ha implementat, emprant el llenguatge VHDL, l'autòmat descrit pel diagrama d'estats de la figura 8.17. Aquest autòmat passa d'un estat a un altre en funció únicament del senyal de rellotge de la FPGA. Quan passa un cert nombre de períodes de rellotge (funció de l'estat en el que es troba el autòmat) l'autòmat canvia d'estat. A continuació es descriuen els estats del autòmat:

1. **Estat “Data_Reset”**: S'inicialitzen els registres del sistema a '0' de forma asíncrona. La senyal “Data_Reset” val '0' únicament en aquest estat.

2. **Estat “Scan_In”** : Es transmet de la FPGA al xip la cadena de bits corresponent a la selecció de *challenge* i a la selecció dels BR-PUFs en funcionament. Els bits d'informació es transmeten pel pin “Scan_In”. En aquest estat, el senyal “Scan_In_Clk” segueix la senyal de rellotge de la FPGA.

3. **Estat “Data_Reg”**: Es transmet la cadena de bits dels registres *flip-flop* als registres *latch*. En aquest estat es produeix un pols en el senyal “Data_Reg”.

4. **Estat “Run_PUF”**: S'habilita el funcionament dels BR-PUFs. El senyal “Run_PUF” val '1' en aquest estat. L'autòmat es manté en aquest estat un nombre variable de períodes de rellotge, per poder variar el temps que un PUF està en funcionament.

5. **Estat “State_Reg”**: S'enregistra l'estat dels inversors. En aquest estat es produeix un pols en el senyal “State_Reg”.

6. **Estat “State_Reg_To_Flip_Flop”**: Es transmet l'estat dels inversors dels registre *latch* als registres *flip-flop*. En aquest estat, el senyal “Scan_Out_Clk” segueix el senyal de rellotge de la FPGA.

7. **Estat “Scan_Out_Select”**: Estat intermedi per a canviar el mode de funcionament dels multiplexors.

8. **Estat "Scan_Out"** : S'extreuen els estats dels inversors de forma seriada a través del terminal "Scan_Out". En aquest estat, la senyal "Scan_Out_Clk" segueix la senyal de rellotge de la FPGA.

La taula 8.1 mostra els senyals en funció de l'estat en el qual es troba l'autòmat. El codi VHDL del autòmat de control es troba a l'annex F.

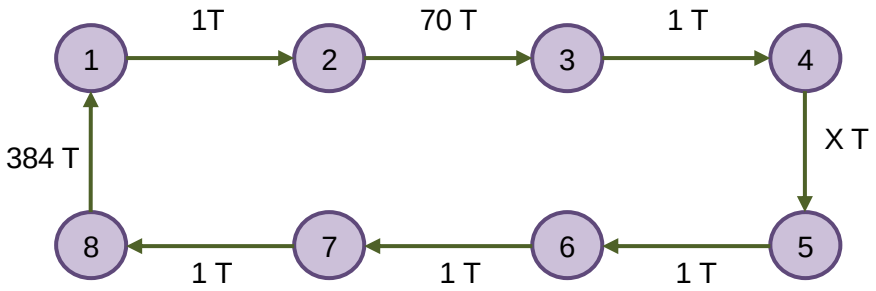


Figura 8.17: Diagrama d'estats del autòmat de control

| Estat | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Scan_In | X | IN | X | X | X | X | X | X |
| Scan_In_Clk | 0 | CLK | 0 | 0 | 0 | 0 | 0 | 0 |
| Data_Reg | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| Run_PUF | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| State_Reg | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| Scan_Out_Select | X | X | X | X | 0 | 0 | 1 | 1 |
| Data_Reset | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Scan_Out_Clk | 0 | 0 | 0 | 0 | 0 | CLK | 0 | CLK |
| Scan_Out | 0 | 0 | 0 | 0 | 0 | 0 | 0 | OUT |

Taula 8.1: Senyals de control en funció de l'estat de l'autòmat. IN simbolitza la transmissió de dades de la FPGA al xip. OUT simbolitza la extracció dels estats dels inversors. CLK significa que el senyal segueix el senyal de rellotge de la FPGA.

Per poder programar una seqüència de PUFs a avaluar, l'autòmat de control també actua sobre una memòria ROM interna a la FPGA. La memòria ROM té una capacitat per a guardar 256 paraules de 72 bits. Cada paraula de la memòria guarda el conjunt format pels 64 bits *challenge* i 6 bits de selecció de BR-PUF en funcionament. A causa de que la FPGA no pot implementar memòries amb paraules de 70 bits, hi haurà 2 bits de cada paraula que no s'utilitzaran. La FPGA recorre de forma seqüencial les paraules de la memòria ROM per a poder avaluar varis *challenges* de forma seqüencial. La figura 8.18 presenta el funcionament d'aquest sistema.

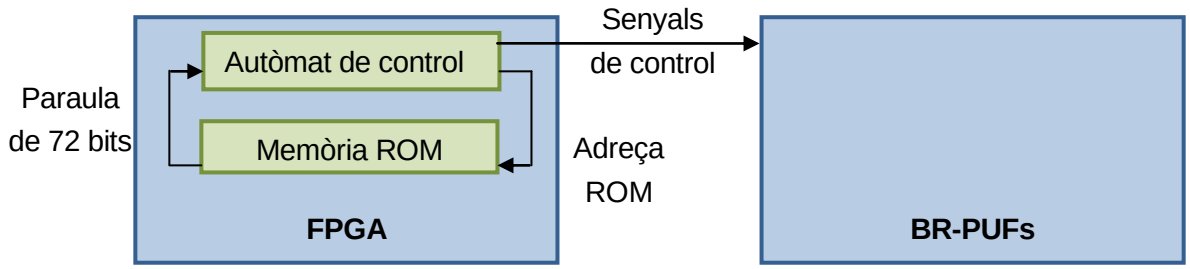


Figura 8.18: Funcionament del sistema format per l'autòmat de control i la memòria ROM

### 8.4.3. Inicialització ROM

La memòria ROM s'inicialitza mitjançant un fitxer tipus .hex que conté, en format hexadecimal, el valor de les paraules amb les que s'inicialitzarà la ROM. Els 64 bits menys significatius, es a dir, els 16 primers dígits hexadecimals, corresponen als bits de *challenge*. Els 2 dígits hexadecimals més significatius corresponen als 6 bits de selecció del BR-PUF en funcionament. Com que els dos bits més significatius d'aquesta seqüència no s'utilitzen, els dos dígits corresponents a la selecció dels BR-PUFs poden variar de '00' a '3F'. La taula 8.2 mostra la correlació entre els dos dígits hexadecimals més significatius de les paraules emmagatzemades a la memòria ROM i els BR-PUFs que s'habiliten per al funcionament.

| Hexadecimal | BIT | BR-PUFs en funcionament | | | | | |
|---|---|---|---|---|---|---|---|
| | | 1 (0fF) | 2 (44fF) | 3 (50fF) | 4 (60fF) | 5 (70fF) | 6 (80fF) |
| 00 | 00000000 | - | - | - | - | - | - |
| 01 | 00000001 | x | - | - | - | - | - |
| 02 | 00000010 | - | x | - | - | - | - |
| 03 | 00000011 | x | x | - | - | - | - |
| 04 | 00000100 | - | - | x | - | - | - |
| 05 | 00000101 | x | - | x | - | - | - |
| 06 | 00000110 | - | x | x | - | - | - |
| 07 | 00000111 | x | x | x | - | - | - |
| 08 | 00001000 | - | - | - | x | - | - |
| 09 | 00001001 | x | - | - | x | - | - |
| 0A | 00001010 | - | x | - | x | - | - |
| 0B | 00001011 | x | x | - | x | - | - |
| 0C | 00001100 | - | - | x | x | - | - |
| … | … | … | … | … | … | … | … |
| 3D | 00111101 | x | - | x | x | x | x |
| 3E | 00111110 | - | x | x | x | x | x |
| 3F | 00111111 | x | x | x | x | x | x |

Figura 8.2: PUFs habilitats en funció del primer byte de cada paraula emmagatzemada a la ROM de la FPGA. La taula complerta es troba a l'annex G

# 9. Resultats de les simulacions del BR-PUF

En aquest capítol es comprova el funcionament del BR-PUF mitjançant simulacions SPICE. Es comprovar el funcionament de la interfície de control simulant el codi VHDL del autòmat de control amb el programa Modelsim d'Altera. Finalment s'analitza la *response* obtinguda de la simulació de BR-PUFs mitjançant la utilització de l'entorn de simulació creat amb el llenguatge de programació Python.

## 9.1. Validació del funcionament del sistema

S'ha validat el comportament del sistema format per la FPGA i el circuit implementat en un ASIC de forma separada.

### 9.1.1. Simulació del BR-PUF

Utilitzant el programari Cadence Virtuoso s'han realitzat simulacions SPICE per a validar el funcionament del circuit electrònic implementat.

En primer lloc s'han realitzat simulacions Monte Carlo sobre 50 instàncies de BR-PUFs. La figura 9.1 mostra el transitori d'aquestes simulacions. Comparant el resultat obtingut per al cas de BR-PUF sense filtre IOCF amb el cas del BR-PUF amb filtre IOCF de 80fF és pot observar que les oscil·lacions és cancel·len completament amb la utilització del filtre IOCF.

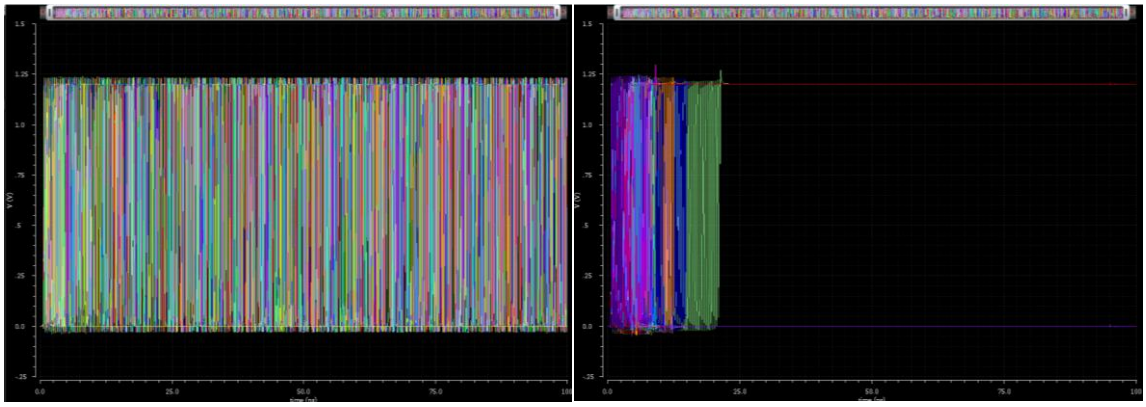

Figura 9.1: Simulació del BR_PUF. Transitori de 50 simulacions Monte Carlo realitzades sobre BR-PUFs de 64 portes NOR amb filtres IOCF de 0 i 80 fF.

Per a comprovar el funcionament del circuit, s'ha simulat un cicle de introducció de *challenge*, posada en marxa del PUF, obtenció i extracció de la *response*. Les simulacions s'han agilitzat emprant únicament una cel·la "BR_PUF_SLICE", es a dir, s'han format anells biestables de 4 portes NOR.

La figura 9.2 mostra el funcionament del detector d'estabilitat. Es pot veure com aquest sistema detecta correctament el final de les oscil·lacions ja que el senyal de sortida del detector (senyal verd) canvia de valor a l'aturar-se les oscil·lacions del anell biestable (senyal vermell).

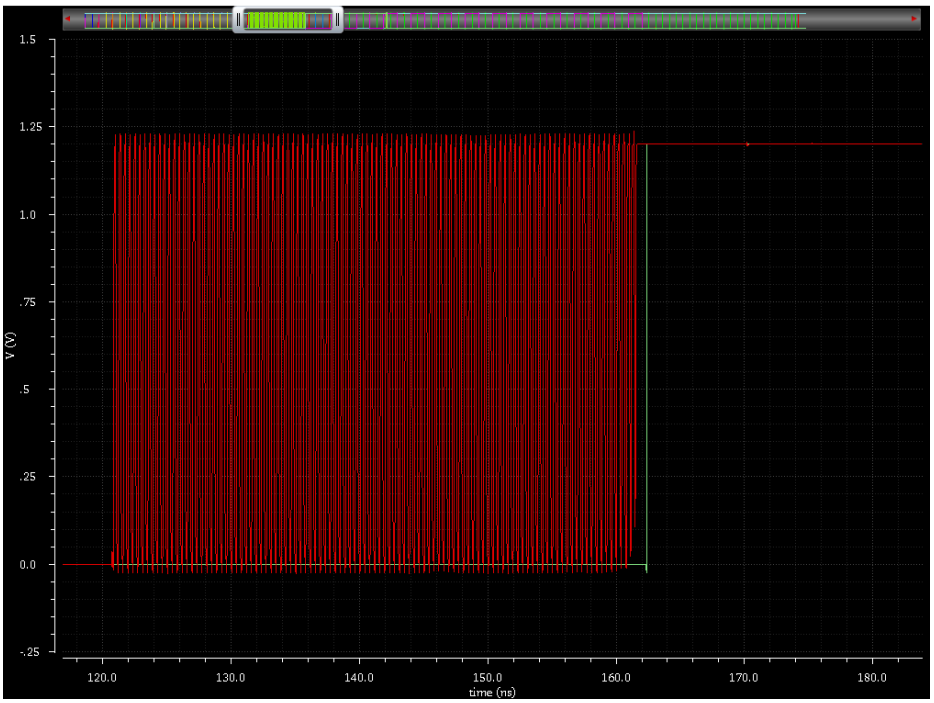


Figura 9.3: Simulació del BR_PUF. Funcionament del detector d'estabilitat. El senyal vermell correspon a l'estat d'un inversor de l'anell biestable. El senyal verd correspon a la sortida del detector d'estabilitat.

La figura 9.4 mostra la senyal de sortida del circuit pel terminal "Scan_Out" (senyal lila). El senyal verd correspon al senyal de control "Scan_Out_Clk". Cada cop que es produeix un flanc positiu en la senyal de rellotge, es transmet un bit d'informació pel terminal "Scan_Out". Aquesta cadena de bits es correspon amb els estats enregistrats de les portes NOR del BR-PUF. Es pot observar que el bit d'informació transmès varia cada flanc positiu de la senyal de rellotge. A més es transmeten 24 bits d'informació corresponents al les 4 portes NOR del cada un dels 6 anells biestables.

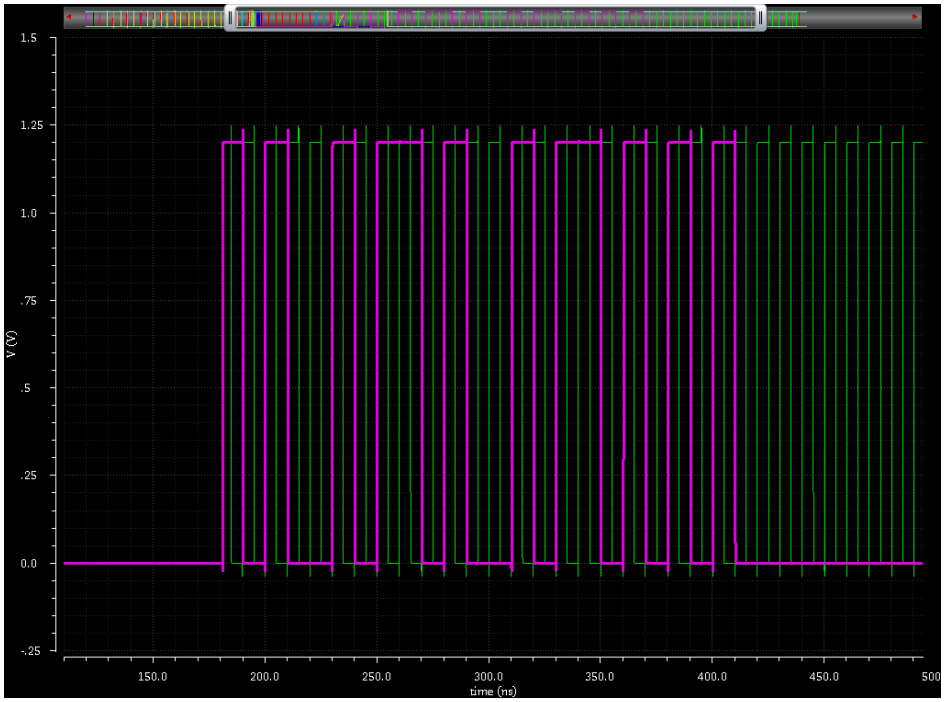


Figura 9.4: Simulació del BR_PUF. Extracció de las estats dels inversors enregistrats. La senyal verda correspon a la senyal “Scan_Out_Clk”. El senyal lila és la cadena de bits que s'emet pel pin “Scan_Out”

### 9.1.2. Simulació de la nterfície de control

Mitjançant el programari Modelsim d’Altera s'ha simulat el comportament de l’autòmat de control. La figura 9.5 mostra l’evolució dels senyals de control del BR-PUF durant un cicle complet de funcionament del circuit. Inicialment s’inicialitzen els registres del circuit amb un pols negatiu del senyal “Data_Reset”. Seguidament es transmet una cadena de bits pel terminal “Scan_In” alhora que el senyal de rellotge “Scan_In_Clk” produeix cops de rellotge. A continuació es posen en funcionament els PUFs posant a nivell alt la senyal “Run_PUF”. Finalment s’enregistren els estats del BR-PUF i s’activa la senyal de rellotge “Scan_Out_Clk” per a extreure la *response* del BR-PUF.

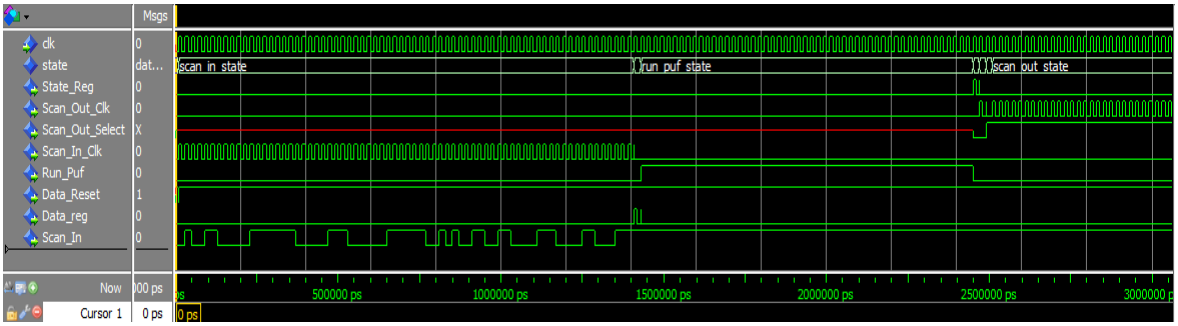


Figura 9.5: Simulació de la interfície de control

## 9.2. Anàlisi de la *response* dels BR-PUFs implementats

Per tal de caracteritzar els BR-PUFs implementats s'han simulat 50 instàncies les quals se'ls hi han aplicat els mateixos 128 *challenges*. La simulació s'ha dut a terme amb el model descrit per la equació 6.12, amb els paràmetres calculats a l'apartat 8.2.2.1. Mitjançant les equacions 4.2, 4.5, i 4.6 s'han analitzat les *responsees* obtingudes en els BR-PUFs simulats. Cal destacar que la fiabilitat del PUF no s'ha calculat ja que el model no inclou la possibilitat d'afegir variacions generades per pertorbacions externes.

La taula 9.1 mostra la caracterització dels BR-PUFs analitzats per a diferents valors del filtre IOCF. S'observa que l'aplicació del filtre no afecta de forma significativa a les propietats del PUF. La unicitat és del 50% i la *inter-distance* segueix una distribució gaussiana com es mostra a la figura 9.6. Aquest fet permet, en teoria, crear entitats de BR-PUF diferenciables. A més el *bit-aliasing* segueix una distribució gaussiana centrada al 50% (figura 9.7), mostra de que aquest sistema no presenta el problema del *bit-aliasing*.

| PUF | Cf (fF) | Unicitat | Desviació estàndard *inter-distance* | Mitja Uniformitat | Desviació estàndard Uniformitat | Mitja *Bit-Aliasing* | Desviació estàndard *Bit-Aliasing* |
|---|---|---|---|---|---|---|---|
| 64 *Challenge* Bits BR-PUF NOR | 44 | 50,31 | 0,181 | 51,66 | 29,76 | 51,66 | 5,66 |
| | 50 | 50,57 | 0,188 | 48,84 | 30,41 | 48,84 | 4,57 |
| | 60 | 50,18 | 0,208 | 53,09 | 31,88 | 53,09 | 5,59 |
| | 70 | 49,98 | 0,148 | 53,84 | 25,95 | 53,84 | 6 |
| | 80 | 50,25 | 0,159 | 48,7 | 27,55 | 48,7 | 5,99 |

Taula 9.1: Caracterització de BR-PUFs de 64 bits de *challenge* amb filtres IOCF de diferent capacitat

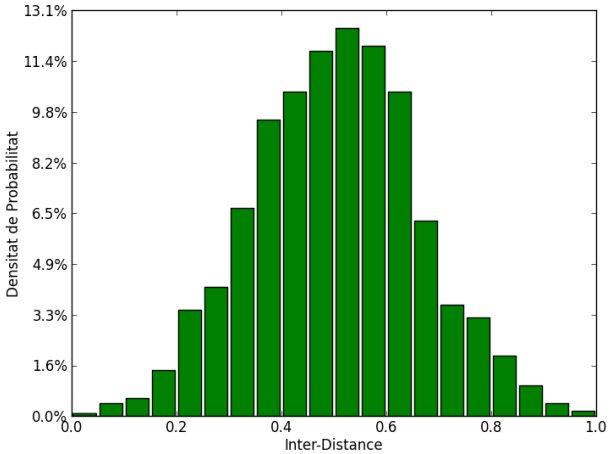


Figura 9.6: Histograma de les distàncies de les *inter-distances* entre les *responses* dels BR-PUFs simulats amb filtre IOCF de 80 fF

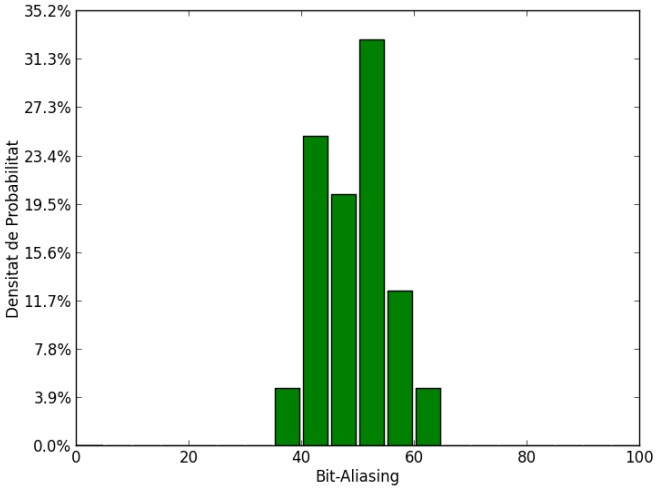


Figura 9.7: Histograma del *bit-aliasing* dels BR-PUFs simulats amb filtre IOCF de 80 fF

El problema de la distribució aleatòria a la uniformitat es manté ja que la desviació estàndard d'aquest paràmetre és molt elevada, fent que aquest paràmetre segueixi una distribució aleatòria (figura 9.8). Com ja s'ha estudiat al capítol 7, mitjançant la implementació del sistema TBR-PUF s'aconseguiria convertir aquesta distribució aleatòria en una distribució gaussiana sense modificar la resta de característiques del BR-PUF.

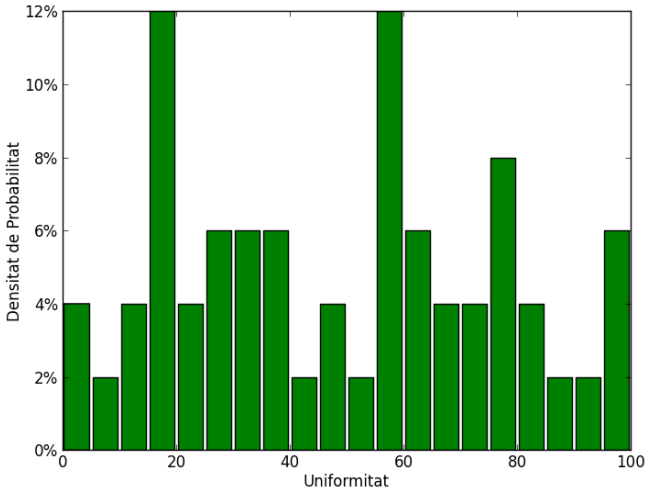


Figura 9.8: Histograma de les uniformitats dels BR-PUFs simulats amb filtre IOCF de 80 fF

# Conclusions

En aquest projecte s'ha presentat una arquitectura de PUF anomenada BR-PUF. S'ha analitzat aquest sistema utilitzant diferents metodologies i s'han identificat diversos problemes. Primerament, observant el transitori de les simulacions realitzades, s'ha detectat que aquest sistema tendeix a oscil·lar de forma incontrolada i impredictible. Aquest fet provoca que a la pràctica el PUF tingui una resposta lenta a l'hora obtenir un conjunt de parelles *challenge/response* significativament gran que per a identificar el PUF correctament. A més, analitzant la *response* del PUF, s'ha observat una distribució aleatòria en la uniformitat de les instàncies de BR-PUF simulades, fet que pot posar en compromís la seguretat del dispositiu.

La supreció de les oscil·lacions s'ha assolit dissenyant i dimensionant un filtre, anomenat IOCF, que redueix de forma dràstica el temps d'estabilització. Per a anells biestables de 32 inversors sense filtre IOCF s'ha observat que nomes un 55% de les instancies simulades s'havien estabilitzat al finalitzar la simulació (100ns). En canvi, a l'afegir un filtre IOCF de 50fF totes les simulacions s'han estabilitzat als 20ns.

L'arquitectura TBR-PUF mostrada permet solucionar la uniformitat aleatòria dels PUFs, convertint-la en una distribució gaussiana centrada al 50%. A més s'ha observat que la utilització conjunta del TBR-PUF i del filtre IOCF permet solucionar els dos problemes del BR-PUF.

S'ha realitzat el disseny d'un prototip del circuit per a ser implementat en un ASIC. Aquest circuit està format per 6 BR-PUFs amb filtres IOCF de diferentes capacitats. L'objectiu d'aquest prototip és el d'observar com afecta el filtre IOCF al comportament del BR-PUF. S'ha dissenyat un autòmat de control per poder controlar el prototip de forma externa. Aquest autòmat s'ha descrit amb el llenguatge VHDL i està implementat en una FPGA.

El darrer pendent de realitzar, i que es durà a terme en un proper *run* de la tecnologia, és la implementació física d'aquest disseny en un ASIC. Amb això es podrà validar el funcionament experimental del circuit i caracteritzar el BR-PUF amb filtre IOCF. Utilitzant l'autòmat de control dissenyat es podrà programar l'obtenció de CRPs de forma seqüencial.

# Agraïments

Voldria agrair al professor Salvador Manich la seva implicació i dedicació en la realització d'aquest projecte de final de carrera. Sense la seva inestimable ajuda, aquest projecte no s'hagués pogut realitzar. Des del primer instant m'ha donat suport en tasques tant acadèmiques com administratives. A més estic especialment agraït per l'esforç que ha realitzat perquè pogués realitzar una estada al institut AISEC Fraunhofer treballant en l'àmbit dels PUFs.

També vull agrair a l'enginyer Robert Hesselbarth, que durant la meva estada a l'institut AISEC em va ajudar a comprendre el funcionament dels PUFs, i em va donar suport en la realització del model matemàtic del BR-PUF amb filtre IOCF. També vull agrair-li que em permetés utilitzar i ampliar el seu entorn de simulació de BR-PUFs programat amb el llenguatge de programació Python.

Finalment m'agradaria agrair al tècnic Juan Ramón Trilla el facilitar-me i configurar-me les eines de treball que he requerit per a dur a terme aquest projecte. Tot i l'elevat nombre de problemes que han sorgit durant la realització d'aquest, en Juan Ramón sempre ha estat a la meva disposició per a solucionar-los en el menor temps possible.

# Bibliografia.

## Referències bibliogràfiques

**[1]** CHEN,Q. [et al]. *The Bistable Ring PUF: A New Architecture for Strong Physical Unclonable Functions*. HOST: IEEE Computer Society, 2011, p.134-141.

**[2]** CHEN,Q. [et al]. *Characterization of the Bistable Ring PUF*. *Design, Automation & Test in Europe Conference & Exhibition*, 2012, p. 1459-1462.

**[3]** HESSELBARTH, R. *Modeling and Improving Bistable Ring PUFs*. Munich, 2014.

**[4]** MANICH, S., COBOS, M. *Oscillation Canceller Filter for IR-PUFs*. Barcelona, 2013.

**[5]** LAWSON,N. *Side-Channel Attacks on Cryptographic Software. IEEE Security & Privacy*. Vol. 7, 2009, p. 65-98.

**[6]** BARENGHI, A. [et al]. *Fault Injection Attacks on Cryptographic Devices: Theory, Practice, and Countermeasuresk. Proceedings of the IEEE*. Vol.10, 2012, p. 3056-3076.

**[7]** MAES, R. VERBAUWHEDE, I. *Physically Unclonable Functions: A Study on the State of the Art and Future Research Directions*. *Towards Hardware-Intrinsic Security. Foundations and Practice*, 2010, p 3-37.

**[8]** RAVIKANTH, P. *Physical One-Way Functions*. MIT, 2001.

**[9]** LIM, D. *Extracting Secret Keys from Integrated Circuits*. Cambridge, 2004.

**[10]** LOFSTROM, K. DAASCH, W. TAYLOR, D. *IC identification circuit using device mismatch*. IEEE Intl. Solid-State Circuit Conference, 2000.

**[11]** SUH, G. DEVEDAS, S. *Physical Unclonable Functions for Device Authentication and Secret Key Generation. Design Automation Conference*, 2007, p. 9-14.

**[12]** DEJEAN, G. KIROVSKI, D. *RF-DNA: Radio-Frequency Certificates of Authenticity*. *Cryptographic Hardware and Embedded Systems*, 2007, p. 346-363.

**[13]** CHEN, Q. [et al]. *Analog circuits for physical cryptography*. *IEEE Intl. Symp. of Integrated Circuits*, 2009, p. 121-124.

**[14]** TUYLS, P. [et al]. *Read-Proof Hardware from Protective Coatings. Cryptographic Hardware and Embedded Systems*, 2006, p. 369-383.

**[15]** HELINSKI, R. ACHARYYA, D. PLUSQUELLIC, J. *A Physical Unclonable Function Defined Using Power Distribution System Equivalent Resistance Variations*. *Annual Design Automation Conference* 2009, p. 647-681.

**[16]** FRUHASHI, K. SHIOZAKI, M. AKITAKA, T. *The Arbiter-PUF with High Uniqueness utilizaing Novel Arbiter Circuit with Delay-Time Measurement*. *Circuit and Systems,* 2011, p. 2325-2328.

**[17]** RÜHRMAIR, U. *Modeling attacks on physical unclonable functions. Conference on Computer and Communicatios Security*, 2010 p. 237-249.

**[18]** EIORA, S. BATURONE, I. *Circuit Authentication based on Ring-Oscillator PUFs*. *Electronics, Circuits ans Systems*, 2011, p. 691-694.

**[19]** MAITI, A. [et al]. *A large scale characterization of RO-PUF*. *IEEE Intl. Symp on Harware-Oriented Security and Trust*, 2010, p. 94-99.

**[20]** HOLOCOMB, D. BULESON, W. FU,K. *Initial SRAM State as a Fingerprint and Source of True Random Numbers for RFID Tags*. *Conference on RFID Security*, Malaga, 2007.

**[21]** KUMAR, S. [et al]. *The Butterfly PUF Protecting IP on every FPGA*. *Workshop on Hardware-Oriented Security and Trust*, 2008, p. 67-70.

**[22]** CLEMENS, C. DMITRY, J. *Cloning Physically Unclonable Functions. Hardware-Oriented Security and Trust*, 2013, p. 1-6.

**[23]** ARMKNECHT, F. [et al]. *A Formal Foundation for the Security of Physical Functions.* IEEE Symposium on Security and Privacy, 2011.

**[24]** MAITI, A. GUNREDDY, V. SCHAUMONT,P. *A Systematic Method to Evaluate and Compare the Performance of Physical Unclonable Functons*. Blacksburg, 2013.

**[25]** HESSELBARTH, R. *Modeling Bistable Inverter Rings*. Munich, 2012.

## Bibliografia complementària

- WESTE, N. *Principles of CMOS VLSI design: a system perspective*. Reading, 1993. Disseny de circuits integrats utilitzant la tecnologia CMOS.

- ALTERA. *Cyclone III Device Handbook*. San Jose, 2011. Manual d'usuari de la FPGA Cyclone III.
- TERASIC. *DE0 User Manual. Development and Education Board*. 2009. Manual d'usuari de la placa DE0.
- STMICROELECTRONICS. *CMOS065 Bulk Design Rules Manual*. 2010. Regles de disseny de la tecnologia CMOS de 65 nm.
- STMICROELECTRONICS. *CMOS065_AMS Design Rules Manual*. 2010. Regles de disseny de la tecnologia CMOS de 65 nm per a senyals analògiques.

# A. Pressupost

El cost del projecte es divideix en les següent partides:

| Concepte | Preu unitari | Unitats | Preu |
|---|---|---|---|
| Fabricació del prototip per la empresa CMP STMicroelectronics | 11500€/$mm^2$ | 0,344$mm^2$ | 3956€ |
| Placa de desenvolupament Altera DE0 de Terasic | 87€ | 1 | 87€ |
| Estació de treball amortitzada a 6 anys | - | - | 640€ |
| Llicència d'un any del programari Cadence per a universitats | 600€ | 1 | 600€ |
| Eines ofimàtiques per a la redacció del treball amortitzades a 6 anys | - | - | 100€ |
| Direcció del PFC | 100 €/hora | 20 setmanes, 2 hores/setama | 4000€ |
| Salari d'enginyer superior segons "XVII Convenio Colectivo de Empresas de Ingeniería y Oficinas de Estudios Técnicos (BOE 25-10-2013)" | 1687,02€/mes | 5 mesos | 8435,10€ |
| | | | |
| **TOTAL** | | | 17818,10€ |

# B.Plec de Condicions

## B.1. Fabricació del BR-PUF

El circuit integrat amb el prototip dissenyat s'ha de fabricar mitjançant la tecnologia CMOS de 65nm. Aquest chip s'ha de fabricar a través de l'organització CMP (Circuits Multi-Projects) que s'encarrega d'organitzar la producció, en volums reduïts, de prototips de circuit integrats per a universitats i laboratoris de recerca. Per reduir costos, aquesta organització transmet el disseny del prototip, juntament amb d'altres dissenys, a l'empresa STMicroelectronics, que és la encarregada de la fabricació del prototip.

## B.2. Utilització de la FPGA

Per implementar l'autòmat de control del prototip s'ha d'utilitzar la placa de desenvolupament DE0 de Terasic. Aquesta placa conté, entre altres components, la FPGA Cyclone III de Altera. Per a connectar la FPGA amb el prototip s'han d'utilitzar els ports d'entrada i sortida de propòsit general que incorpora la placa DE0. La figura B.1 mostra una imatge de la placa DE0.

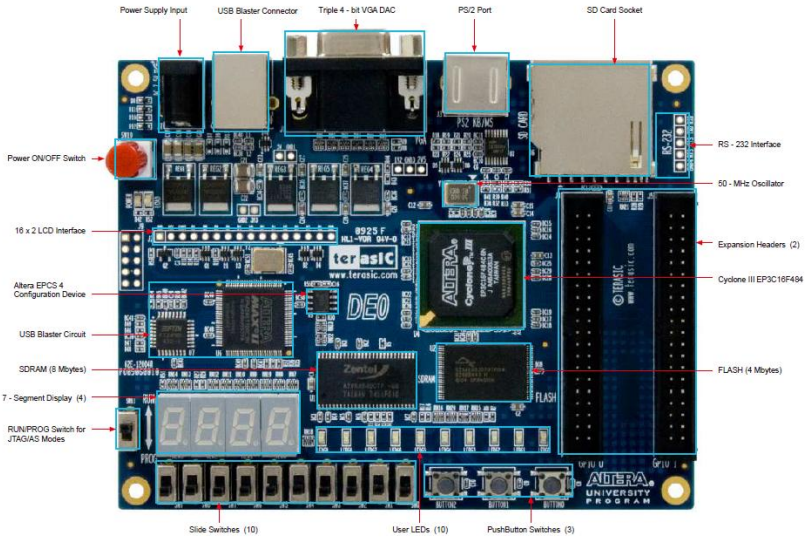


Figura B.1: Placa DE0 de Terasic. Figura extreta del manual d'usuari de la placa DE0

La placa DE0 s'ha de connectar amb el prototip seguint la correlació de pins que mostra la taula B.1. L'estàndard de tensió a utilitzar per aquests pins és el de 1,2V. La figura B.2 mostra la distribució dels pins d'entrada i sortida a la placa DE0.

| Pin del prototip | Pin de la placa DE0 | Pin de la FPGA |
|---|---|---|
| Scan_In | GPIO0_D2 | PIN_AA15 |
| Scan_In_Clk | GPIO0_D7 | PIN_AA13 |
| Data_Reg | GPIO0_D4 | PIN_AA14 |
| Run_PUF | GPIO0_D1 | PIN_AA16 |
| State_Reg | GPIO0_D0 | PIN_AB16 |
| Scan_Out_Select | GPIO0_D6 | PIN_AB13 |
| Data_Reset | GPIO0_D3 | PIN_AB15 |
| Scan_Out_Clk | GPIO0_D8 | PIN_AB10 |
| Scan_Out | GPIO0_D5 | PIN_AB14 |

Taula B.1: Correlació entre els pins del xip prototip, pins de la placa DE0 i pins de la FPGA

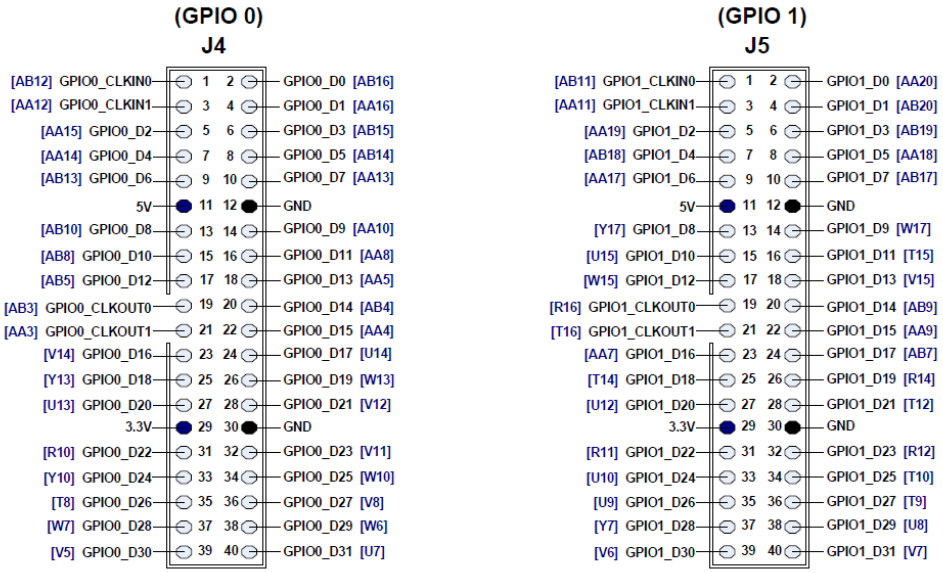


Figura B.2: Ports d'entrada i sortida generals de la placa DE0. En blau es mostra el pin de la FPGA associat. Figura extreta del manual d'usuari de la placa DE0

Per a seleccionar la senyal de rellotge de l'autòmat de control hi ha 2 opcions:

- Es pot utilitzar la senyal de rellotge interna de la placa DE0 de 50 MHz. S'ha d'assignar la senyal de rellotge al pin G21. L'estàndard de tensió a utilitzar per aquest pin és el de 2,5V.
- Es pot utilitzar una senyal de rellotge externa a la placa DE0. Aquesta senyal s'ha de connectar al port GPIO0_CLKIN0 de la placa. S'ha d'assignar la senyal de rellotge del autòmat al pin AB12.

# C.Estudi d'impacte ambiental

## C.1. Fabricació dels components a utilitzar en l'experiment

El fabricant del prototip, STMicroelectronics, i el fabricant de la placa DE0, Terasic, garanteixen que els seus productes estan fabricats complint la directica RoHS (*Restriction of Hazardous Substances*). Com el seu nom indica, la directiva RoHS restringeix la utilització de certs materials contaminants en la fabricació de equips elèctrics i electrònics. Aquesta directiva fixa les següents concentracions màximes de material:

- 0.1% per plom, mercuri, crom VI, PBB (bifenil polibromat) i PBDE (èter difenílic polibromat) del pes en materials homogenis.
- 0.01% per cadmi del pes de material homogeni.

Mitjançant el compliment d'aquesta normativa es minimitza l'impacte ambiental produït en la fabricació o utilització de components electrònics.

## C.2. Reciclatge i recuperació de components electrònics

La directiva WEEE (*Waste Electrical and Electronic Equipment*) s'encarrega de promoure la reutilització, reciclatge i recuperació de residus equips electrònics. L'objectiu d'aquesta directiva és el d'evitar que els components tòxics presents en components electrònics arribin al medi ambient. A més. Mitjançant la reutilització d'aquests components es pretén allargar el cicle de vida d'aquests productes.

# D.Esquemàtics de les cel·les que formen el prototip

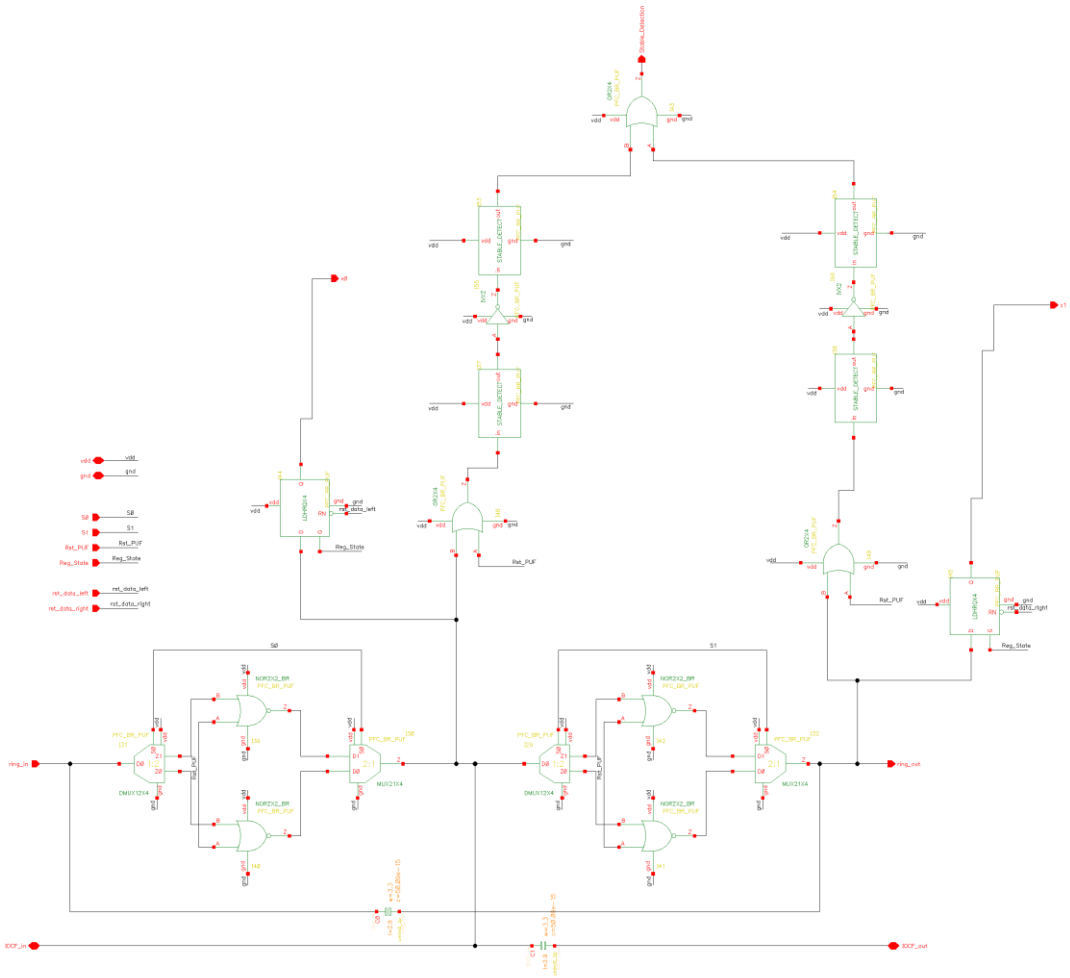

Figura D.1: Esquemàtic de la cel·la "BR_PUF_STAGE"

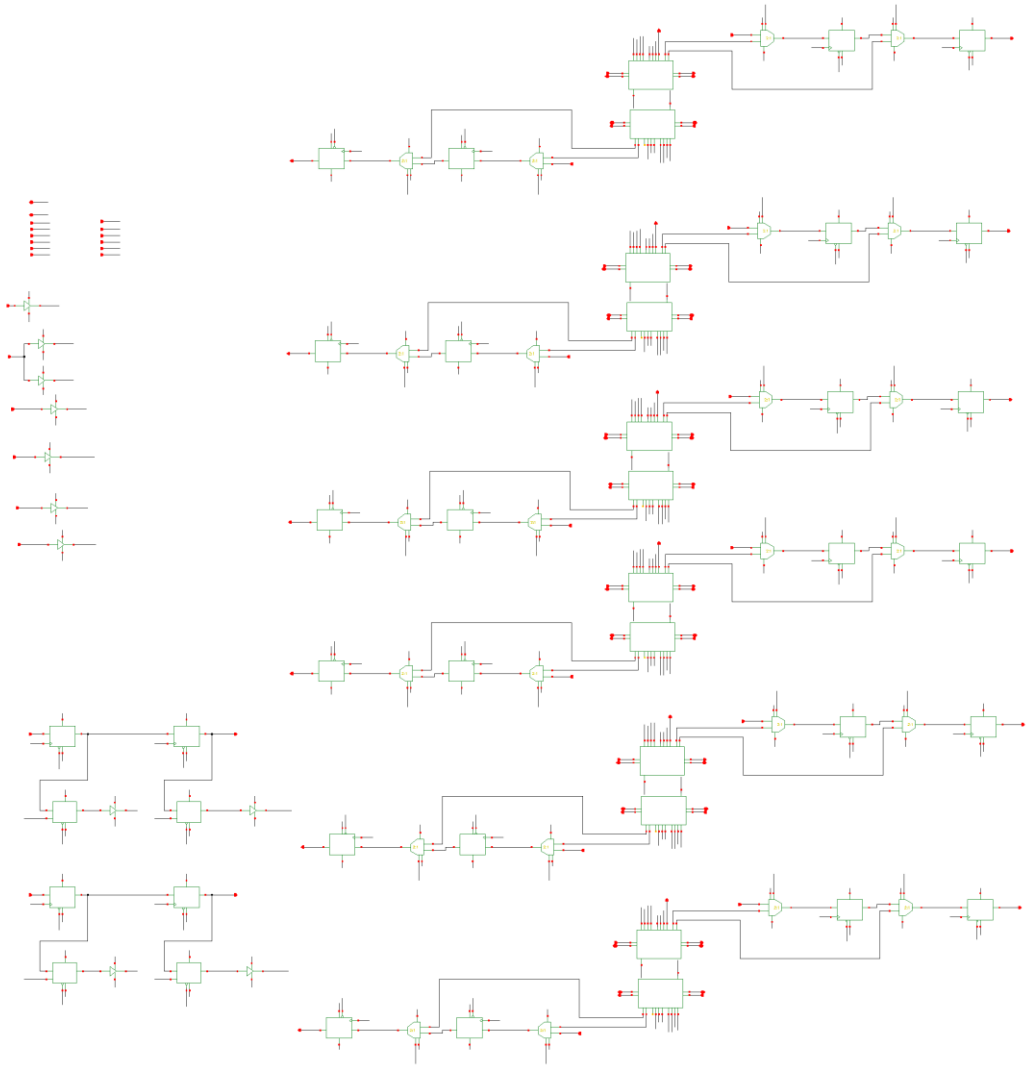

Figura D.2: Esquemàtic de la cel·la "BR_PUF_SLICE"

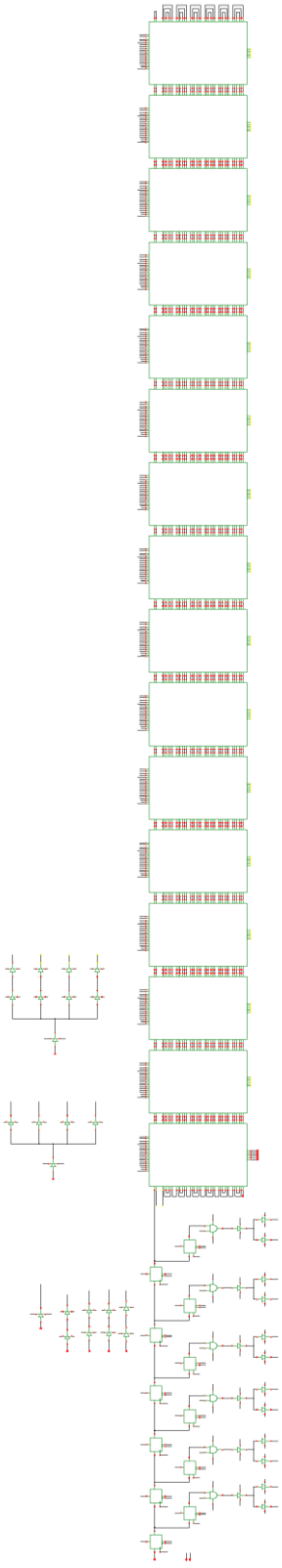

Figura D.3: Esquemàtic de la cel·la "BR_PUF"

# E.*Layout* de les cel·les que formen el prototip

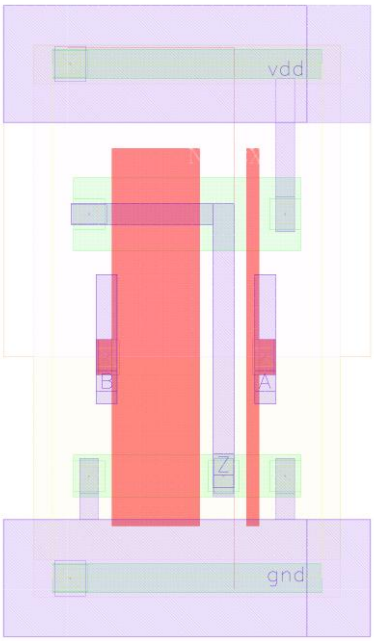


Figura E.1: *Layout* de la cel·la "NOR_BR" porta NOR utilitzada en l'anell biestable

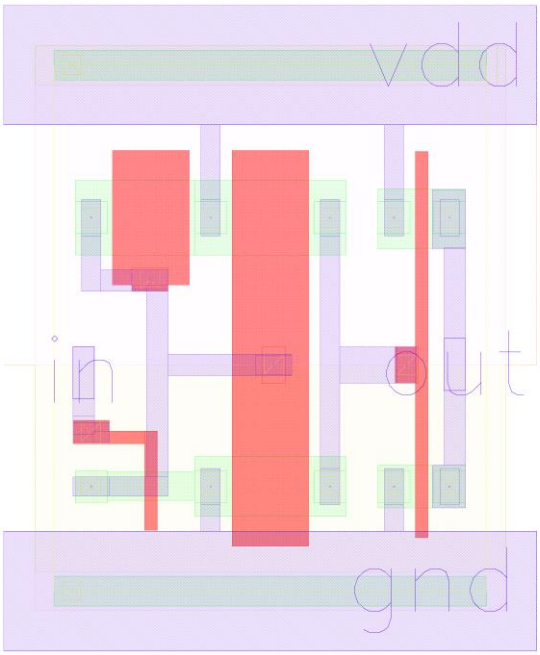


Figura E.2: *Layout* del detector d'estabilitat

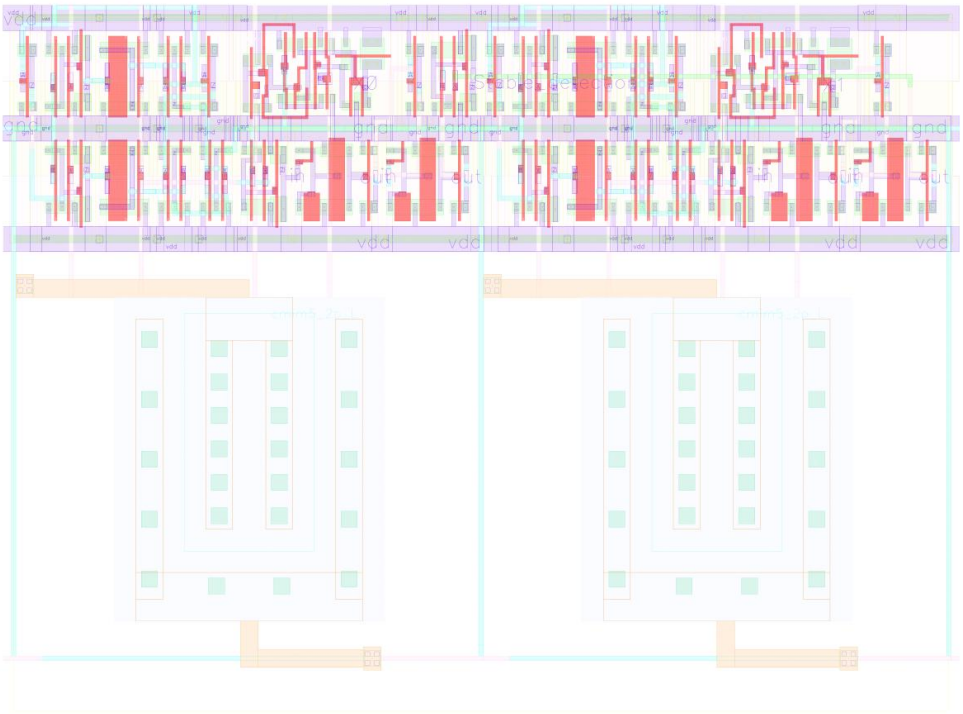

Figura E.3: Layout de la cel·la "BR_PUF_STAGE"

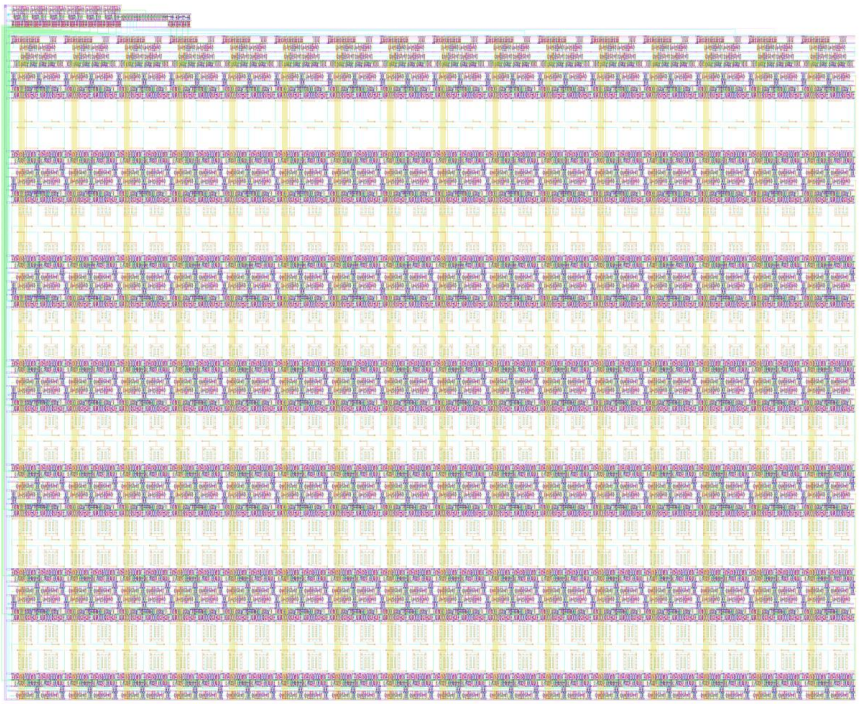

Figura E.4: *Layout* de la cel·la "BR_PUF"

# F. Codi VHDL del autòmat de control

## F.1. Input_Output_Management

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity input_output_management is
  port
  (
    clk                : in std_logic;
    Scan_In_Data       : in std_logic_vector (71 downto 0);


    State_Reg, Scan_Out_Select, Run_PUF, Data_Reset, Data_Reg : out
     std_logic := '0';
    Scan_In_clk        : out std_logic := '0';
    Scan_Out_clk       : out std_logic := '0';

    Data_Adress_Reg    : out std_logic := '0';
    Data_Adress        : buffer integer range 255 downto 0 := 0;
    Scan_In            : out std_logic := '0'

   );
end input_output_management;


architecture funcional of input_output_management is

constant cnt_time_limit: integer := 50;
constant cnt_scan_in_limit: integer := 69;   -- 64 bits challenge +
6 bits selecció PUF
constant cnt_scan_out_limit: integer := 383; -- 64 bits state * 6
PUFs
type automat is(data_reset_state, scan_in_state, data_reg_state,
run_puf_state, state_reg_state, state_reg_to_flip_flop_state,
scan_out_select_state, scan_out_state);
signal state     : automat := data_reset_state;
signal cnt       : integer := 0;


begin


      --automat
      automat_control : process (clk)
      begin
            if clk'event and clk = '1' then
             case state is
```

```
                when data_reset_state =>
                        state <= scan_in_state;

                when scan_in_state =>
                        Scan_In <= Scan_In_Data (cnt);
                        if cnt < cnt_scan_in_limit then
                                cnt <= cnt + 1;
                        else
                                cnt <= 0;
                                state <= data_reg_state;
                        end if;

                when data_reg_state =>
                        state <= run_puf_state;

                when run_puf_state =>
                        if cnt < cnt_time_limit then
                                cnt <= cnt + 1;
                        else
                                cnt <= 0;
                                state <= state_reg_state;
                        end if;

                when state_reg_state =>
                        state <= state_reg_to_flip_flop_state;

                when state_reg_to_flip_flop_state =>
                        state <= scan_out_select_state;

                when scan_out_select_state =>
                        state <= scan_out_state;

                when scan_out_state =>
                        if cnt < cnt_scan_out_limit then
                                cnt <= cnt + 1;
                        else
                                cnt <= 0;
                                Data_Adress <= Data_Adress + 1;
                                state <= data_reset_state;
                        end if;

            end case;
        end if;
    end process automata_control;


State_Reg   <= clk when state = state_reg_state else
               '0';
```

```
Scan_Out_Select  <= '1' when state = scan_out_state or state =
                         scan_out_select_state else
                    '0' when state = state_reg_state or state =
                         state_reg_to_flip_flop_state else
                    'X';

Run_PUF          <= '1' when state = run_puf_state or state =
                         state_reg_state else
                    '0';

Data_Reset       <= not (clk) when state = data_reset_state else
                    '1';

Data_Reg         <= clk when state = data_reg_state else
                    '0';

Scan_In_clk      <= clk when state = scan_in_state else
                    '0';

Scan_Out_clk     <= clk  when state = state_reg_to_flip_flop_state
                         or state = scan_out_state else
                    '0';

Data_Adress_Reg  <= clk  when state = data_reset_state else
                    '0';

end funcional;
```

## F.2. Memòria ROM

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

LIBRARY altera_mf;
USE altera_mf.all;

ENTITY rom IS
      PORT
      (
            address          : IN STD_LOGIC_VECTOR (7 DOWNTO 0);
            inclock          : IN STD_LOGIC  := '1';
            q          : OUT STD_LOGIC_VECTOR (71 DOWNTO 0)
      );
END rom;

ARCHITECTURE SYN OF rom IS

      SIGNAL sub_wire0 : STD_LOGIC_VECTOR (71 DOWNTO 0);


      COMPONENT altsyncram
```

```
	GENERIC (
		address_aclr_a		: STRING;
		clock_enable_input_a		: STRING;
		clock_enable_output_a		: STRING;
		init_file		: STRING;
		intended_device_family		: STRING;
		lpm_hint		: STRING;
		lpm_type		: STRING;
		numwords_a		: NATURAL;
		operation_mode		: STRING;
		outdata_aclr_a		: STRING;
		outdata_reg_a		: STRING;
		widthad_a		: NATURAL;
		width_a		: NATURAL;
		width_byteena_a		: NATURAL
	);
	PORT (
			address_a	: IN STD_LOGIC_VECTOR (7 DOWNTO 0);
			clock0		: IN STD_LOGIC ;
			q_a	: OUT STD_LOGIC_VECTOR (71 DOWNTO 0)
	);
	END COMPONENT;

BEGIN
	q    <= sub_wire0(71 DOWNTO 0);

	altsyncram_component : altsyncram
	GENERIC MAP (
		address_aclr_a => "NONE",
		clock_enable_input_a => "BYPASS",
		clock_enable_output_a => "BYPASS",
		init_file => "Scan_In_Data.hex",
		intended_device_family => "Cyclone III",
		lpm_hint => "ENABLE_RUNTIME_MOD=NO",
		lpm_type => "altsyncram",
		numwords_a => 256,
		operation_mode => "ROM",
		outdata_aclr_a => "NONE",
		outdata_r  eg_a => "UNREGISTERED",
		widthad_a => 8,
		width_a => 72,
		width_byteena_a => 1
	)
	PORT MAP (
		address_a => address,
		clock0 => inclock,
		q_a => sub_wire0
	);


END SYN;
```

# G. Taula d'inicialització de la ROM de la FPGA per a la selecció dels PUFs en funcionament

| Hexadecimal | BIT | BR-PUFs en funcionament | | | | | |
|---|---|---|---|---|---|---|---|
| | | 1 (0fF) | 2 (44fF) | 3 (50fF) | 4 (60fF) | 5 (70fF) | 6 (80fF) |
| 00 | 00000000 | - | - | - | - | - | - |
| 01 | 00000001 | x | - | - | - | - | - |
| 02 | 00000010 | - | x | - | - | - | - |
| 03 | 00000011 | x | x | - | - | - | - |
| 04 | 00000100 | - | - | x | - | - | - |
| 05 | 00000101 | x | - | x | - | - | - |
| 06 | 00000110 | - | x | x | - | - | - |
| 07 | 00000111 | x | x | x | - | - | - |
| 08 | 00001000 | - | - | - | x | - | - |
| 09 | 00001001 | x | - | - | x | - | - |
| 0A | 00001010 | - | x | - | x | - | - |
| 0B | 00001011 | x | x | - | x | - | - |
| 0C | 00001100 | - | - | x | x | - | - |
| 0D | 00001101 | x | - | x | x | - | - |
| 0E | 00001110 | - | x | x | x | - | - |
| 0F | 00001111 | x | x | x | x | - | - |
| 10 | 00010000 | - | - | - | - | x | - |
| 11 | 00010001 | x | - | - | - | x | - |
| 12 | 00010010 | - | x | - | - | x | - |
| 13 | 00010011 | x | x | - | - | x | - |
| 14 | 00010100 | - | - | x | - | x | - |
| 15 | 00010101 | x | - | x | - | x | - |
| 16 | 00010110 | - | x | x | - | x | - |
| 17 | 00010111 | x | x | x | - | x | - |
| 18 | 00011000 | - | - | - | x | x | - |
| 19 | 00011001 | x | - | - | x | x | - |
| 1A | 00011010 | - | x | - | x | x | - |
| 1B | 00011011 | x | x | - | x | x | - |
| 1C | 00011100 | - | - | x | x | x | - |
| 1D | 00011101 | x | - | x | x | x | - |
| 1E | 00011110 | - | x | x | x | x | - |
| 1F | 00011111 | x | x | x | x | x | - |
| 20 | 00100000 | - | - | - | - | - | x |
| 21 | 00100001 | x | - | - | - | - | x |
| 22 | 00100010 | - | x | - | - | - | x |
| 23 | 00100011 | x | x | - | - | - | x |
| 24 | 00100100 | - | - | x | - | - | x |
| 25 | 00100101 | x | - | x | - | - | x |
| 26 | 00100110 | - | x | x | - | - | x |
| 27 | 00100111 | x | x | x | - | - | x |
| 28 | 00101000 | - | - | - | x | - | x |
| 29 | 00101001 | x | - | - | x | - | x |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 2A | 00101010 | - | x | - | x | - | x |
| 2B | 00101011 | x | x | - | x | - | x |
| 2C | 00101100 | - | - | x | x | - | x |
| 2D | 00101101 | x | - | x | x | - | x |
| 2E | 00101110 | - | x | x | x | - | x |
| 2F | 00101111 | x | x | x | x | - | x |
| 30 | 00110000 | - | - | - | - | x | x |
| 31 | 00110001 | x | - | - | - | x | x |
| 32 | 00110010 | - | x | - | - | x | x |
| 33 | 00110011 | x | x | - | - | x | x |
| 34 | 00110100 | - | - | x | - | x | x |
| 35 | 00110101 | x | - | x | - | x | x |
| 36 | 00110110 | - | x | x | - | x | x |
| 37 | 00110111 | x | x | x | - | x | x |
| 38 | 00111000 | - | - | - | x | x | x |
| 39 | 00111001 | x | - | - | x | x | x |
| 3A | 00111010 | - | x | - | x | x | x |
| 3B | 00111011 | x | x | - | x | x | x |
| 3C | 00111100 | - | - | x | x | x | x |
| 3D | 00111101 | x | - | x | x | x | x |
| 3E | 00111110 | - | x | x | x | x | x |
| 3F | 00111111 | x | x | x | x | x | x |

Taula G.1: PUFs habilitats en funció del primer byte de cada paraula emmagatzemada a la ROM de la FPGA