UNIVERSITÀ
DEGLI STUDI
DI PADOVA

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA IN INGEGNERIA ELETTRONICA

FPGA Kintex UltraScale per applicazioni spaziali

Relatore:
Daniele Vogrig

Laureando:
Edoardo Vergerio

Data di laurea 21/09/2022

Anno Accademico 2021/2022

Indice

1 Introduzione

L'obiettivo di questo elaborato di tesi è quello di esplorare le caratteristiche più significative degli FPGA Kintex UltraScale di Xilinx utilizzati in applicazioni spaziali.
Per avere un'iniziale prospettiva, si propone una breve discussione riguardo le possibili applicazioni e il mercato in cui questi dispositivi potrebbero essere utilizzati.
Si passa poi ad esplorare le caratteristiche base di un FPGA, riportando alcune soluzioni, implementate dall'architettura UltraScale, per aumentare le prestazioni e per facilitare il loro impiego in applicazioni spaziali.
In seguito, viene proposta una caratterizzazione dell'ambiente in cui gli FPGA Kintex UltraScale sono chiamati ad operare quando utilizzati in missioni spaziali e vengono descritte le tipologie di radiazione a cui sono sottoposti.
Si analizzano i principali effetti che le radiazioni hanno su qualsiasi dispositivo elettronico, inclusi gli FPGA, e di cui è necessario tenere conto durante la fase di progettazione.
Infine, vengono riportate alcune delle principali tecniche di mitigazione, ovvero soluzioni hardware & software che è possibile implementare a bordo degli FPGA Kintex UltraScale per contrastare gli effetti delle radiazioni e per garantire l'operatività del dispositivo nelle difficili condizioni in cui esso opera.

1.1 Xilinx: profilo aziendale

Xilinx è una multinazionale che progetta e produce una vasta gamma di dispositivi logici programmabili e strumenti di sviluppo software. Essa è maggiormente nota per essere stata la prima azienda ad aver introdotto nel mercato gli FPGA, ovvero Field Programmable Gate Array. Fondata nella Silicon Valley nel 1984 da Ross Freeman, l'azienda è oggi proprietà di AMD (Advanced Micro Devices). Il focus aziendale è rivolto ad offrire dispositivi altamente personalizzabili dall'utente finale, in modo che quest'ultimo possa ottenere performance e adattabilità anche nelle applicazioni più specifiche. I settori in cui Xilinx è attualmente impegnata vengono riportati in Figura 1.

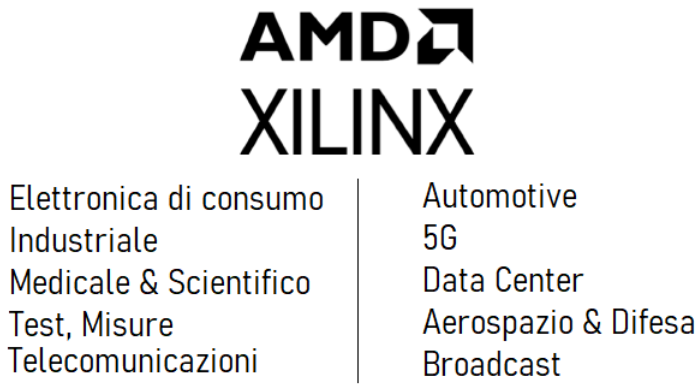

Figura 1: Settori di interesse Xilinx. [1]

1.2 Gli FPGA Kintex Ultrascale

La famiglia di FPGA Kintex UltraScale di Xilinx si propone come strumento per permettere al settore spaziale di raggiungere capacità di calcolo dell'ordine delle centinaia di Gbit/s. Ciò rende possibile un notevole passo avanti in tutte le applicazioni di tipo spaziale, non solo fornendo migliori performance di calcolo, ma anche aumentando le potenzialità nei trasferimenti tra dispositivi di I/O, di memoria interna e soprattutto dando la possibilità di operare la riconfigurazione del dispositivo direttamente in orbita.
Tra le principali applicazioni che potrebbero usufruire di tali caratteristiche vi sono, ad esempio, l'osservazione in tempo reale (real-time streaming) della terra dallo spazio, l'utilizzo di sensori ad alta risoluzione, lo space-based Internet e telecomunicazioni mobili a banda larga con la possibilità di ottimizzare/scalare il sistema utilizzato, secondo le necessità del momento, sfruttando la capacità di riconfigurazione.
Tradizionalmente infatti, l'elaborazione di segnali nello spazio è sempre stata eseguita utilizzando dispositivi ASIC – Application Specific Integrated Circuit, che seppur estremamente efficienti per svolgere l'attività prevista, non hanno la possibilità di essere riconfigurati, operazione che invece potrebbe rendersi necessaria nel caso in cui i protocolli utilizzati dovessero essere modificati o aggiornati.
Caratteristica fondamentale per applicazioni di tipo spaziale è la robustezza all'effetto delle radiazioni dell'ambiente spaziale. Gli FPGA Kintex UltraScale vengono testati in laboratorio per caratterizzare il grado di radiazione da essi sopportato e per garantire il corretto funzionamento in tali condizioni. Inoltre, per mitigare gli effetti delle radiazioni, tali dispositivi implementano topologie circuitali e tecniche di design provenienti da più di 40 brevetti Xilinx.
In un'indagine di mercato condotta da Xilinx [2], è previsto che a partire dall'anno 2020 più di 1000 satelliti saranno lanciati ogni anno per il prossimo decennio per offrire i servizi prima enunciati. In particolare, la richiesta sarà principalmente legata alla realizzazione di costellazioni di mini satelliti, da lanciare in orbite equatoriali geosincrone (GEO) e orbite terrestri basse (LEO). La vita media di tali dispositivi si aggira attorno ai 3-5 anni se utilizzati in orbite LEO, anche se molte aziende sono in competizione per offrire particolari servizi in orbite superiori, come ad esempio lo space-based internet. [2]

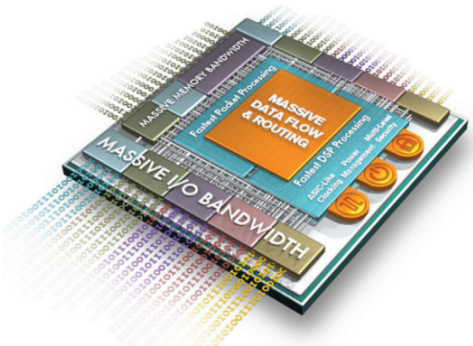

Figura 2: FPGA Kintex UltraScale per applicazioni a banda larga. [3]

1.3 Descrizione generale di un FPGA

Per Field Programmable Gate Array (FPGA) si intende un circuito integrato costituito da una matrice di blocchi logici elementari. Tali blocchi, chiamati CLB, Configurable Logic Block, possono essere configurati da un programmatore per eseguire le più svariate funzioni logiche.
Gli FPGA possono essere riprogrammati molteplici volte; il termine "Field Programmable" evidenzia come le funzionalità del dispositivo non siano stabilite a priori, ma che invece possano essere definite in base alle esigenze del tipo di applicazione in cui viene impiegato. Tuttavia esistono anche versioni del tipo OTP, One Time Programmable, la cui configurazione non può essere modificata successivamente. In Figura 3 è possibile osservare la struttura base della matrice di un FPGA. Ciascun CLB può essere collegato ad altre tipologie di blocco elementare, come ad esempio altri CLB, memorie RAM, interfacce I/O, o altri moduli aventi funzioni specifiche e adatti per alcune particolari applicazioni, come i blocchi DSP.

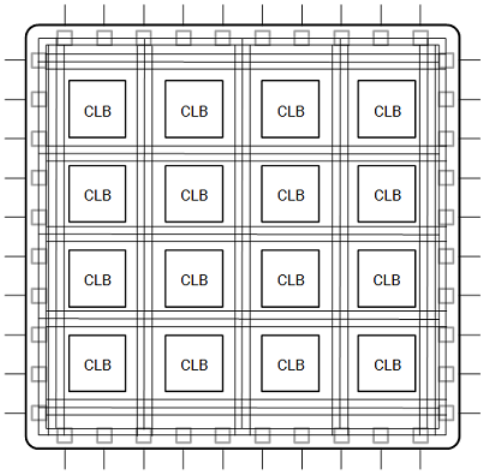

Figura 3: Matrice di blocchi elementari (CLB) e interconnessioni. [4]

La rete di interconnessioni è ampiamente configurabile dal programmatore e viene utilizzata per realizzare strutture circuitali complesse. La realizzazione delle interconnessioni è molto spesso una funzione integrata in specifici ambienti di sviluppo che, grazie all'utilizzo di algoritmi dedicati, consentono di gestire al meglio l'utilizzo delle risorse e massimizzare le performance.
Uno dei metodi per programmare un FPGA è l'utilizzo di un linguaggio di descrizione dell'hardware, come ad esempio il VHDL. Attraverso tale linguaggio è possibile descrivere il circuito di interesse e, usando software di sviluppo appositi, tradurlo in un flusso di bit di configurazione, anche detto *configuration bitstream*.

I bit di configurazione vengono utilizzati per definire le funzioni logiche necessarie alla realizzazione del circuito e per regolare la rete di interconnessioni tra i vari blocchi elementari che compongono l'FPGA. Tali bit sono custoditi all'interno di una memoria dedicata chiamata Configuration Memory (CRAM).
In Figura 4 viene riportato un esempio di codice VHDL usato per descrivere un multiplexer a 3 ingressi.

```vhdl
library IEEE;
use IEEE.std_logic_1164.all;

entity Mux3 is
    generic (k : integer := 1);
    port(a0,a1,a2 : in std_logic_vector(k-1 downto 0);
         s : in std_logic_vector(2 downto 0);
         b : out std_logic_vector(k-1 downto 0));
end Mux3;

architecture implCase of Mux3 is
begin
    process(all) begin
        case s is
            when "001" => b <= a0;
            when "010" => b <= a1;
            when "100" => b <= a2;
            when others => b <= (others => '-');
        end case;
    end process;
end implCase;

architecture implSelect of Mux3 is
begin
    with s select
        b <= a0 when "001",
             a1 when "010",
             a2 when "100",
             (others => '-') when others;
end implSelect;
```

Figura 4: Esempio di codice VHDL usato per descrivere un multiplexer a 3 ingressi.

Con la crescita delle dimensioni e della complessità degli FPGA, per facilitare il processo di sviluppo, Xilinx offre la possibilità di utilizzare dei moduli di funzioni esterni già pronti all'uso e detti IP, *"Intellectual Property"*. Utilizzando un approccio di design a blocchi chiamato *"Block Design"*, disponibile all'interno dell'ambiente di sviluppo Vivado, è possibile progettare dei sistemi complessi interconnettendo tra loro i diversi moduli IP richiesti e senza la necessità di ricorrere a linguaggi di descrizione dell'hardware.
Nello specifico, sfruttando il tool *"IP Integrator"*, l'operazione di posizionamento e interconnessione dei blocchi avviene attraverso l'uso di un'interfaccia grafica (GUI).
In aggiunta, l'IP Integrator offre la funzionalità *"Connection Automation"* per assistere lo sviluppatore nella gestione di tutte le connessioni necessarie al corretto funzionamento dei moduli. Quanto posizionato all'interno dell'interfaccia costituisce un *Block Design* che potrà essere esportato come un pacchetto da riutilizzare in molteplici progetti o che verrà sintetizzato in un *configuration bitstream* per essere implementato nell'FPGA target.
Un esempio di quanto appena descritto viene proposto in Figura 5, dove è possibile osservare i moduli IP posizionati all'interno di una comoda interfaccia grafica. [5]

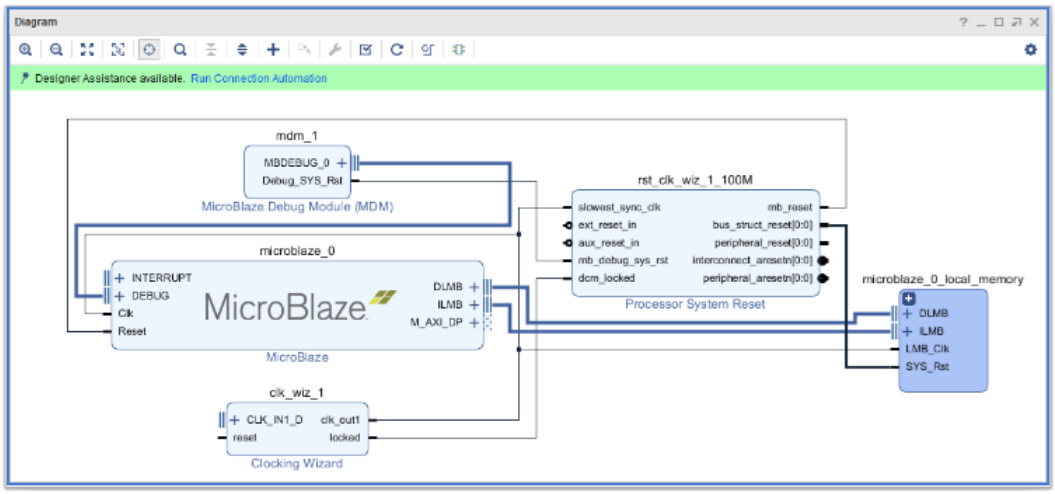

Figura 5: Esempio di utilizzo dell'interfaccia grafica di IP Integrator. [5]

2 Architettura UltraScale

Per soddisfare le richieste di un mercato sempre più esigente in termini di prestazioni, si è resa necessaria la progettazione di una nuova architettura hardware.
Xilinx ha quindi sviluppato l'architettura UltraScale a 20nm, la colonna portante di due famiglie di FPGA ad alte prestazioni: Kintex UltraScale & Virtex Ultrascale.
Entrambe le famiglie condividono la stessa base hardware, ma con una differente combinazione di risorse disponibili a bordo (DSP, CLB, RAM ecc.) e con un fine applicativo diverso. La serie Kintex è ottimizzata per applicazioni di tipo DSP, Digital Signal Processing, ovvero per l'elaborazione digitale dei segnali. In Tabella 1 viene presentata la differente composizione delle risorse non solo tra l'architettura UltraScale e la precedente serie 7, ma anche tra le due diverse famiglie disponibili.

Risorse dispositivo	**Kintex-7**	**Kintex UltraScale**	**Virtex-7**	**Virtex UltraScale**
Celle Logiche	478	1451	1995	5541
Block RAM (Mb)	34	76	68	133
DSP48	1920	5520	3600	2880
Performance DSP (GMACs)	2845	8180	5335	4268
Transceiver	32	64	96	120
Banda Transceiver (Gb/s)	800	2086	2784	5616
Performance interfaccia di memoria (Mb/s)	1866	2400	1866	2400

Tabella 1: Confronto Serie 7 - UltraScale per le famiglie di FPGA Kintex e Virtex. [6]

La struttura elementare precedentemente discussa verrà ora estesa per meglio caratterizzare alcuni degli elementi fondamentali dell'architettura UltraScale, il cui schema a blocchi viene illustrato in Figura 6.

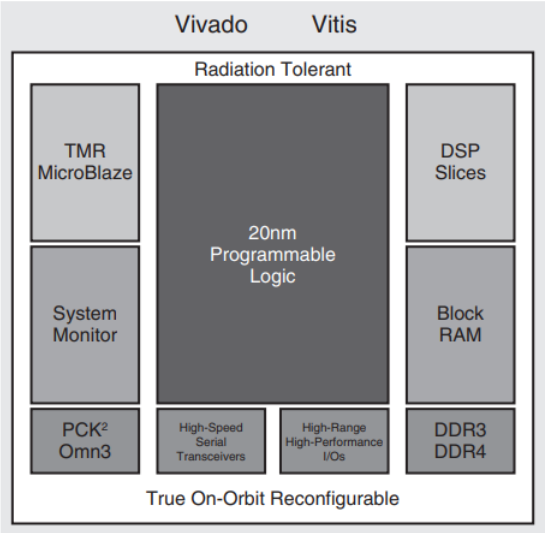

Figura 6: Schema a blocchi della piattaforma UltraScale. [2]

2.1 CLB

L'unità programmabile fondamentale è costituita dal CLB, la risorsa principale per l'implementazione di circuiti logici combinatori e sequenziali. Nell'architettura UltraScale, un CLB è composto da uno *slice*. Ciascuno slice è costituito da 8 LUT (Look-Up Table) a 6 ingressi e da 16 registri Flip-Flop. Gli slice e i CLB associati sono organizzati in colonne verticali all'interno della matrice programmabile. L'interconnessione di più CLB permette di realizzare funzioni logiche complesse. I tool di sviluppo disponibili al programmatore permettono di organizzare i CLB in modo efficiente, riducendo la necessità di interconnessioni lunghe e compattando quindi il design. Tali accorgimenti implementativi sono rappresentati in Figura 7, dove è possibile apprezzare la riduzione delle risorse utilizzate grazie ad una migliore gestione delle interconnessioni. [7]

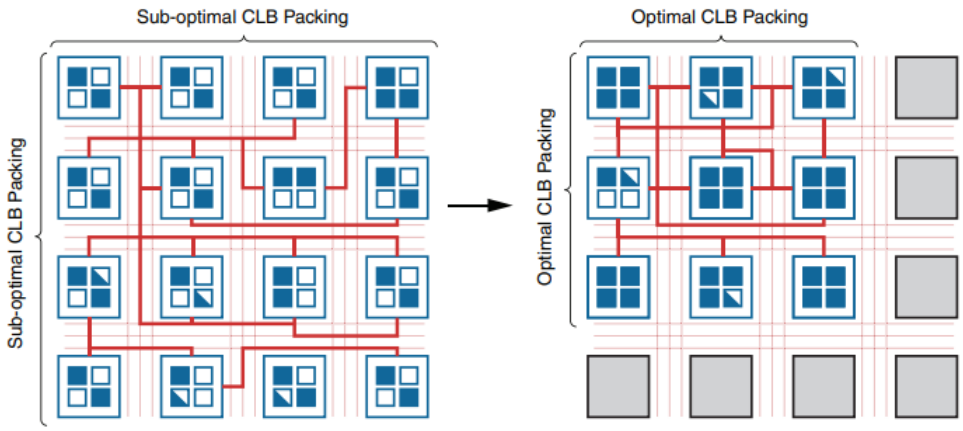

Figura 7: Disposizione efficiente dei CLB e riduzione delle interconnessioni. [6]

In Figura 8 si riporta la struttura di uno slice. Al suo interno sono presenti le LUT a 6 input per ognuna delle quali sono messi a disposizione 2 registri Flip-Flop.

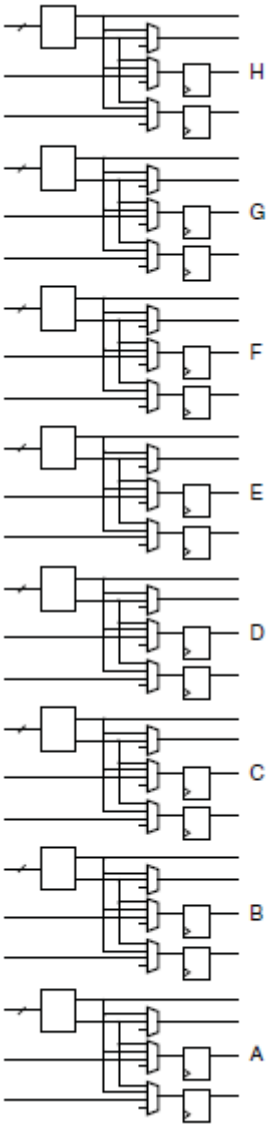

Figura 8: LUT e Flip-Flop all'interno di uno slice. [7]

Nell'architettura UltraScale esistono due differenti tipologie di slice.

- SLICEL: uno slice di tipo L (logica) viene utilizzato per implementare funzioni logiche.
- SLICEM: uno slice di tipo M (memoria) può essere utilizzato anche come memoria aggiuntiva. Sfruttando le LUT è possibile realizzare una memoria distribuita, perché fisicamente posizionata in CLB distinti all'interno della matrice.

2.1.1 LUT

Una LUT (Look-Up Table) è ciò che permette di realizzare una qualunque funzione logica combinatoria. Essa è una sorta di memoria a cui è possibile accedere specificando un certo indirizzo. L'indirizzo corrisponde a una sequenza di bit che vengono forniti come ingressi alla LUT. A ciascun indirizzo è associato un bit che corrisponde all'uscita, il risultato della funzione logica. Nell'architettura UltraScale è possibile realizzare LUT con 6 bit di ingresso e quindi $2^6 = 64$ possibili locazioni indirizzabili. E' possibile utilizzare le LUT anche con una configurazione di ingressi ridotta. Infatti, le LUT a 6 ingressi sono realizzate usando due LUT a 5 ingressi, combinate tra loro da un multiplexer.
In Tabella 2 viene riportato un esempio di LUT a 6 ingressi, mentre in Figura 9 è possibile osservare come i segnali di uscita della LUT possano essere gestiti attraverso multiplexer e Flip-Flop.

Ingressi						*Uscita*
I_0	I_1	I_2	I_3	I_4	I_5	I_O
1	0	1	0	0	0	1
1	0	1	1	0	1	0
1	1	0	0	1	0	1
...						
1	1	1	1	0	0	1

Tabella 2: Esempio di LUT

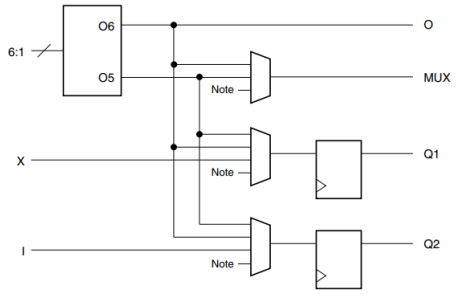

Figura 9: Look-Up Table con 2 Flip-Flop associati. [7]

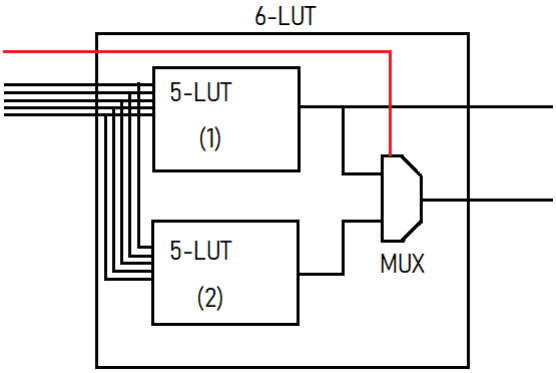

Figura 10: Schema generale di una 6-LUT realizzata con due 5-LUT

In Figura 10 viene riportato lo schema generale per realizzare una LUT a 6 ingressi a partire da due 5-LUT. Per usare la configurazione a 6 ingressi è necessario agire sul multiplexer utilizzando l'ingresso riportato in rosso; per utilizzare le due 5-LUT separatamente, si mantiene fisso il segnale di controllo del multiplexer affinché questo propaghi l'uscita della seconda 5-LUT.

Come precedentemente annunciato, nel caso in cui lo slice sia di tipo M (SLICEM), allora i 64 bit di uscita, corrispondenti alle locazioni indirizzate dagli ingressi, possono essere utilizzati per realizzare una memoria distribuita. Inoltre, collegando più LUT tra loro è possibile realizzare memorie di capacità maggiore e fino ad un massimo di 512 bit per slice (8 LUT per 64 bit di memoria ciascuna). Una 6-LUT può essere configurata in modo da realizzare una memoria 64x1, ma è anche possibile utilizzarla nella configurazione 32x2 sfruttando le due 5-LUT interne, come in Figura 11. In quest'ultima è possibile notare come negli slice di tipo M si renda necessaria l'introduzione di una linea per gli indirizzi di scrittura (WA - Write Address), di un segnale di Write Enable (WE) e di un segnale di clock dedicato. Le linee I & X sono invece usate come ingressi per i dati da inserire in memoria.

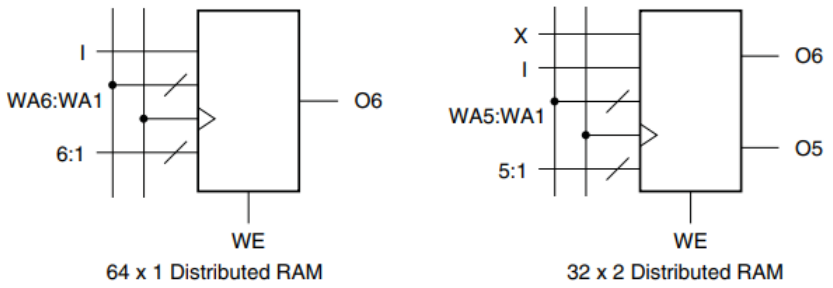

Figura 11: SLICEM nelle configurazioni di memoria distribuita 64x1 & 32x2. [7]

2.1.2 Shift Register

Le LUT degli SLICE di tipo M (SLICEM) possono essere configurate come shift register da 32 bit. In Figura 12 viene riportato lo schema di tale configurazione.

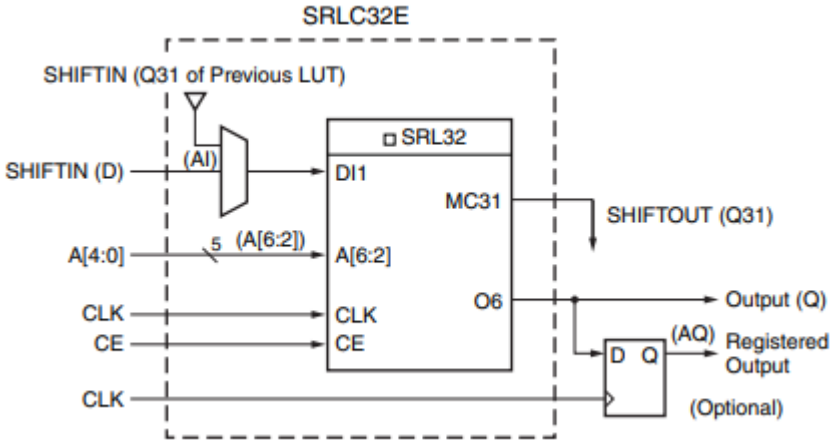

Figura 12: SLICEM configurato come shift register a 32 bit. [7]

Agendo sulla linea degli indirizzi A[4:0] è possibile leggere individualmente i 32 bit dello shift register dal pin di output Q. Tale funzionalità permette la creazione di shift register di dimensioni inferiori, cioè con meno di 32 bit. Ad esempio, volendo creare uno shift register da 20 bit, basterà impostare l'indirizzo al ventesimo bit.
Lo shift register non possiede delle linee dedicate al *SET* né al *RESET*, ma può essere inizializzato secondo le esigenze durante la fase di configurazione.
I pin SHIFTIN & SHIFTOUT(Q31) possono essere usati per concatenare più LUT tra loro, a formare shift register di dimensioni maggiori. All'interno di uno SLICEM infatti, essendo presenti 8 LUT, è possibile realizzare shift register fino a 256 bit. E' infine possibile interconnettere tra loro molteplici SLICEM per creare configurazioni con ancora più bit a disposizione. [7]
Di seguito si riporta una breve descrizione dei pin coinvolti:

- **Data In - DI1**: è l'ingresso (a un bit) dello shift register;
- **Clock Enable - CE**: quando questo pin è a livello logico alto, l'operazione di shift è abilitata;
- **Address - A[4:0]**: 5 bit di ingresso della LUT da usare per selezionare individualmente i 32 bit dello shift register. Il LSB della LUT non è utilizzato e viene posto automaticamente a livello logico alto dai tool di sviluppo;
- **Data Out - Q**: pin di output che riporta il bit selezionato dall'indirizzo;
- **Data Out - Q31**: pin di output che riporta sempre l'ultimo bit dello shift register.

2.1.3 Multiplexer

Ciascuna LUT a sei ingressi può essere impiegata per realizzare dei multiplexer 4:1. Sfruttando la presenza di multiplexer dedicati all'interno dello slice, è possibile combinare più LUT tra loro, in modo da rendere possibile la realizzazione di:

- 8 multiplexer 4:1, usando una LUT ciascuno;
- 4 multiplexer 8:1, combinando due LUT adiacenti;
- 2 multiplexer 16:1, combinando quattro LUT adiacenti;
- 1 multiplexer 32:1, combinando tutte le LUT in uno slice.

La struttura che rende possibile tali implementazioni viene riportata in Figura 13, dove è possibile osservare i multiplexer dedicati F7, F8 e F9 che permettono di combinare le LUT tra loro.

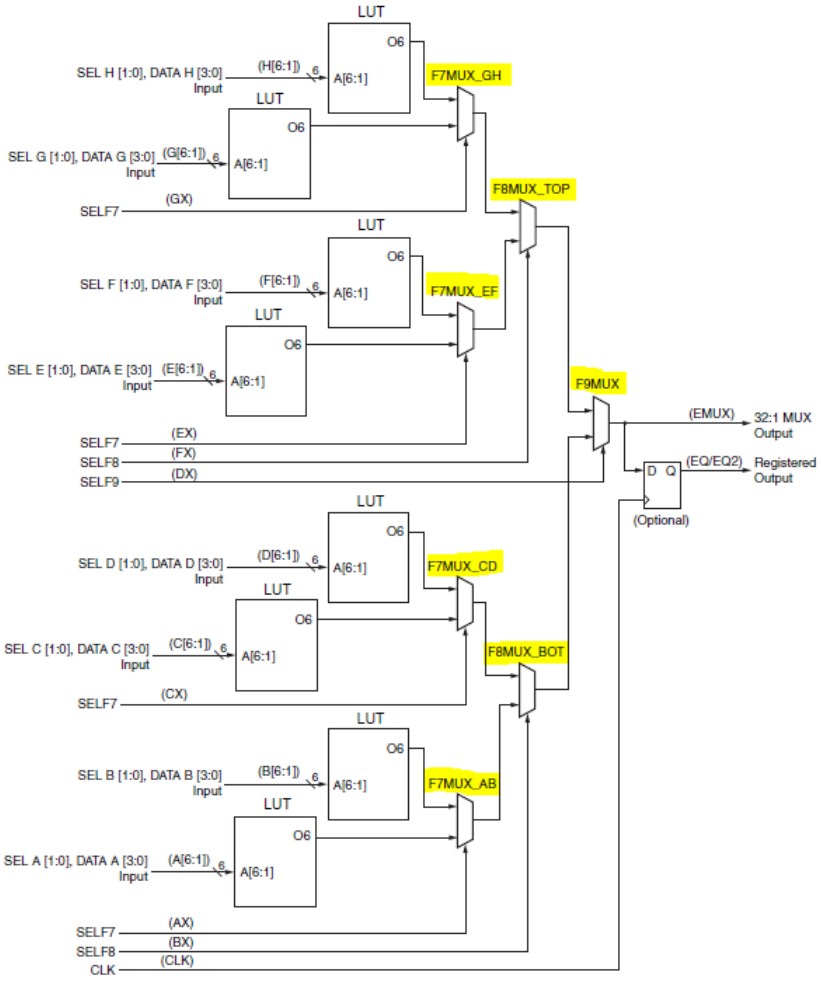

Figura 13: Realizzazione di un MUX 32:1 combinando le LUT di uno slice. In giallo sono evidenziati i multiplexer dedicati che permettono di combinare le LUT. [7]

2.2 Block RAM - BRAM

In aggiunta alla memoria RAM distribuita, l'architettura UltraScale mette a disposizione un elevato numero di blocchi BRAM (Block-RAM). Ciascun blocco BRAM è una memoria da 36 Kbit, costituita internamente da due memorie RAM da 18 Kbit ciascuna.
Un qualsiasi blocco BRAM può essere impiegato in uno dei due seguenti modi:

- due memorie indipendenti da 18 Kbit ciascuna;
- una memoria da 36 Kbit.

Le operazioni di lettura e scrittura possono avvenire contemporaneamente attraverso porte di ingresso e uscita indipendenti. Tra le molteplici configurazioni possibili, in Figura 14 viene riportato il caso di una memoria dual-port sincrona a 36 Kbit.

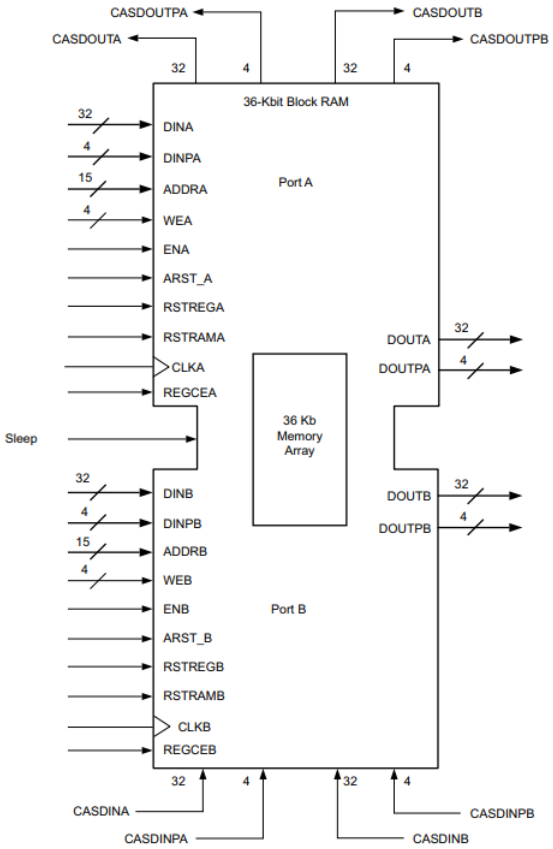

Figura 14: Memoria BRAM dual-port a 36 Kbit. [8]

E' possibile osservare la presenza di due porte, A e B. Ognuna di esse dispone di propri segnali di indirizzo, data in, data out, clock e write enable. Ciò permette di usare le due porte in modo del tutto indipendente sia come ingresso che uscita. Essendo la memoria sincrona, le operazioni di lettura o scrittura ad una porta richiedono la presenza di un fronte del segnale di clock per essere abilitate. Nel caso in cui ad entrambe le porte venga richiesta un'operazione di scrittura allo stesso indirizzo, è necessario arbitrare in modo appropriato i segnali di clock CLKA & CLKB, in modo da scongiurare il verificarsi di conflitti. Nessun conflitto si verifica se invece entrambe le porte richiedono la lettura di uno stesso indirizzo di memoria.

2.3 Blocchi DSP

L'architettura UltraScale permette la realizzazione efficiente di algoritmi DSP - Digital Signal Processing, perché dispone di blocchi elementari appositamente studiati per applicazioni di questo tipo. L'implementazione di algoritmi DSP richiede molto spesso l'utilizzo di moltiplicatori binari e accumulatori, che nell'architettura UltraScale vengono realizzati in strutture dedicate chiamate *DSP Slice*. [9]

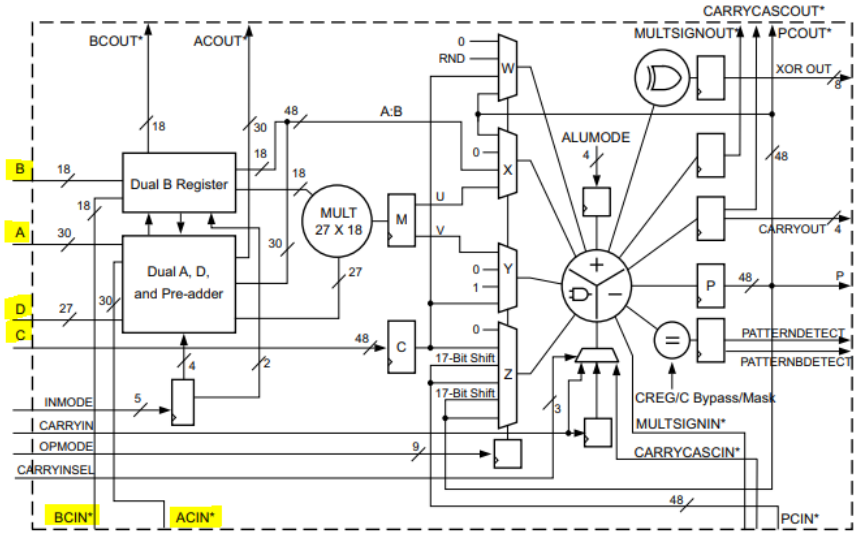

Figura 15: Schema circuitale di un DSP Slice. [9]

In Figura 15 è possibile osservare come un DSP Slice abbia a disposizione quattro canali di ingresso, indicati con A, B, C e D.

Ingresso A	30 bit
Ingresso B	18 bit
Ingresso C	48 bit
Ingresso D	27 bit

Tabella 3: Canali di ingresso DSP Slice

Di seguito vengono riportate alcune operazioni implementabili con tale blocco:

- I 27 bit meno significativi di A e i 18 bit di B possono essere utilizzati come ingressi di un moltiplicatore complemento a due 27x18; è inoltre presente un accumulatore a 48 bit;
- A e B possono essere concatenati per essere utilizzati, insieme all'ingresso C, in un full-adder/sottrattore a 48 bit, utilizzabile anche per eseguire operazioni logiche a 48 bit (AND,OR,XOR, ecc.);
- E' presente un rilevatore di pattern che consente di confrontare l'uscita del blocco DSP con una sequenza nota;
- E' possibile porre in cascata molteplici DSP Slice per realizzare sistemi di pipeline.

2.4 Segnale di Clock

Le architetture precedenti UltraScale erano progettate per avere un sistema di clock che a partire dal centro della matrice dell'FPGA, si diffondeva verso i margini della stessa. Con l'aumento della capacità di calcolo e, più in generale, delle performance totali del dispositivo, il fenomeno del clock skew iniziava ad essere un fattore non più trascurabile. Con clock skew si intende il ritardo tra la generazione del segnale di clock e il suo effettivo arrivo all'elemento circuitale di interesse, ritardo tanto maggiore quanto più l'elemento si trova ai margini del dispositivo.
L'architettura UltraScale si propone come soluzione a questo tipo di problema.
L'FPGA è infatti suddiviso in regioni di clock (CR), ciascuna contenente un certo numero di CLB, DSP e Block RAM. Le regioni sono interconnesse da una fitta rete di collegamenti, di cui è possibile individuare due tipologie:

- Clock routing: rete che indirizza il segnale di clock partendo dalla sorgente globale ed arrivando ai centri delle regioni di clock, anche chiamati *clock root*;
- Clock distribution: rete che distribuisce il segnale di clock partendo dalla *clock root* e raggiungendo i dispositivi contenuti in quella specifica CR.

Il grande vantaggio di tale tipo di organizzazione è che la sorgente del clock, sebbene comune a tutto l'FPGA, può essere posizionata in una specifica regione ed estendersi limitatamente in essa. E' quindi possibile definire delle reti di clock dall'estensione limitata ed in cui gli effetti dello skew siano contenuti. E' inoltre possibile sfruttare la rete di interconnessioni per collegare molteplici CR tra loro. [10]
In Figura 16 è possibile osservare come il segnale di clock venga distribuito nell'architettura UltraScale.

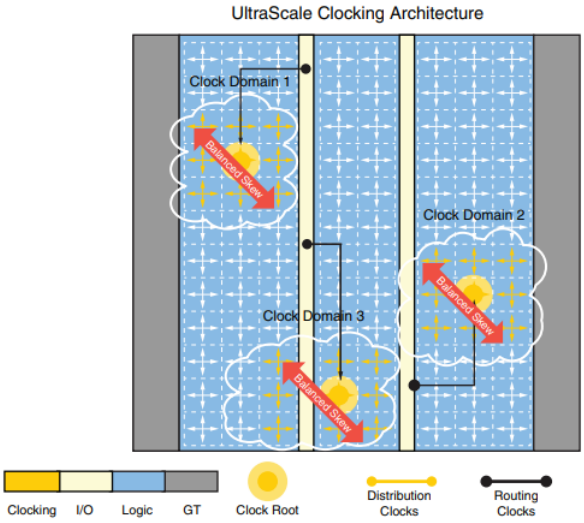

Figura 16: Distribuzione del segnale di clock nell'architettura UltraScale. [10][6]

2.5 Configurazione FPGA

Negli FPGA Kintex UltraScale il bitstream di configurazione viene conservato all'interno di una memoria dedicata chiamata Configuration Memory (CRAM). Tale memoria è realizzata con una tecnologia CMOS altamente robusta e con una buona resistenza agli effetti negativi causati dalle radiazioni dell'ambiente spaziale. La CRAM è una memoria di tipo SRAM. La volatilità di quest'ultima implica che ad ogni accensione essa debba essere configurata nuovamente. Il processo di caricamento del bitstream all'interno della CRAM è detto *configurazione*. Tale fase prevede l'esistenza di una memoria esterna non volatile, detta anche *Golden Memory*, da cui l'FPGA possa prelevare il bitstream da caricare all'interno della CRAM. Alternativamente, il bitstream può essere fornito da altri dispositivi esterni, come ad esempio microcontrollori o PC.
In ogni caso, l'accesso alla memoria di configurazione prevede l'utilizzo di una specifica interfaccia.
A bordo degli FPGA Kintex UltraScale di Xilinx sono disponibili le seguenti interfacce:

- JTAG: interfaccia seriale regolata dallo standard IEEE 1149.1
- SelectMap: interfaccia per il collegamento di periferiche esterne alla memoria di configurazione
- ICAP: interfaccia per l'accesso alla memoria di configurazione da parte di circuiti interni l'FPGA

L'FPGA, attraverso dei circuiti logici dedicati, è in grado di amministrare il processo di configurazione: può procedere alla configurazione dell'intera CRAM oppure di singole regioni. In quest'ultimo caso si parla di riconfigurazione parziale che risulta estremamente utile per aggiornare o introdurre delle nuove funzionalità mantenendo però perfettamente operative le regioni non coinvolte in tale processo. Le regioni di configurazione possono essere classificate in due tipologie:

- Regioni Statiche: contengono i bit di configurazione di elementi critici del sistema e per questo sono escluse dal processo di riconfigurazione.
- Regioni Dinamiche: contengono i bit di configurazione di elementi non critici e quindi possono essere riconfigurate.

Data una qualsiasi regione dinamica, è possibile utilizzare dei moduli di configurazione per rimpiazzare il contenuto precedentemente caricato con quello desiderato.
L'utilizzo della riconfigurazione parziale porta con sé molteplici benefici. Alcuni sistemi infatti non richiedono l'utilizzo di tutti i loro componenti allo stesso momento. Quando un componente non è richiesto è possibile sostituire i bit di configurazione ad esso associati con un modulo per descrivere un componente invece richiesto. Un altro beneficio è quello di poter modificare parte di un circuito per poterne aggiornare le funzionalità senza dover interrompere l'attività delle altre applicazioni.

Infine, la suddivisione dell'FPGA in regioni di configurazione permette una migliore organizzazione e gestione della complessità in sistemi particolarmente estesi. [11][12]

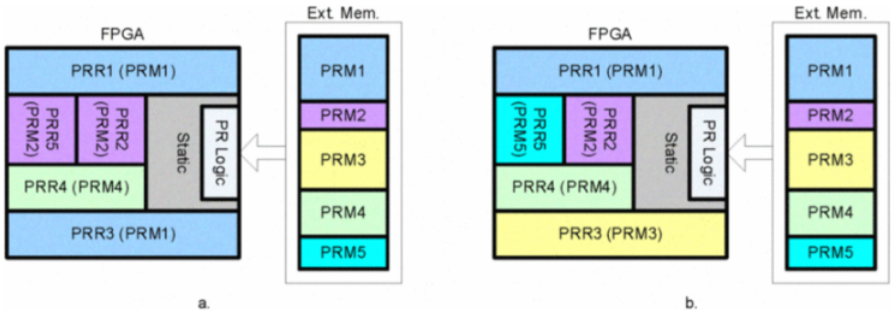

Figura 17: Esempio di riconfigurazione parziale della CRAM. [11]

In Figura 17 è possibile osservare la suddivisione della memoria CRAM in regioni di configurazione distinte in regioni statiche e dinamiche. Grazie al circuito logico di controllo, è possibile eseguire la riconfigurazione parziale. In particolare, nell'immagine proposta, la regione 5 (PRR5) viene riconfigurata sostituendo il modulo PRM2 con il modulo PRM5. Da notare la presenza di una memoria esterna non volatile da cui l'FPGA acquisisce i moduli richiesti.

3 Tolleranza alle Radiazioni

3.1 L'ambiente spaziale - Space Environment

Con ambiente spaziale ci si riferisce a tutte le condizioni a cui un oggetto, ad esempio un circuito elettronico, è esposto durante lo svolgimento di una qualunque missione spaziale. La principale caratteristica di tale ambiente è la presenza di radiazioni ionizzanti, ovvero flussi di energia capaci di ionizzare gli atomi o le molecole dei materiali esposti.
La ionizzazione consiste nel rilascio di uno o più elettroni, evento che può alterare il comportamento di un circuito elettronico.
Le radiazioni sono dette *Direttamente Ionizzanti* se composte da particelle cariche in movimento, come elettroni, protoni o ioni pesanti. Le particelle prive di carica, come i neutroni, e le radiazioni elettromagnetiche ad alta energia, come i raggi X, sono invece definite come radiazioni *Indirettamente Ionizzanti*.
Si riconoscono tre principali sorgenti di radiazioni:

1. Radiazioni della fascia di Van Allen
2. Raggi cosmici
3. Radiazione solare

La fascia di Van Allen è una zona toroidale appartenente alla magnetosfera terrestre al cui interno le cariche trasportate dal vento solare rimangono intrappolate.
Quando delle cariche in movimento si trovano in una regione in cui è presente il campo magnetico terrestre, queste subiscono l'azione della forza di Lorentz e vengono quindi trattenute in un'orbita attorno alla Terra.
A seconda dell'energia delle particelle considerate, è possibile suddividere la fascia di Van Allen in due zone: una interna, costituita da protoni con energia nell'ordine delle centinaia di MeV; una esterna, maggiormente esposta ai venti solari, costituita principalmente da elettroni con energia attorno alle decine di MeV.
La presenza di tale fascia fu scoperta nel 1958 da James Van Allen durante le missioni Explorer 1 ed Explorer 3.

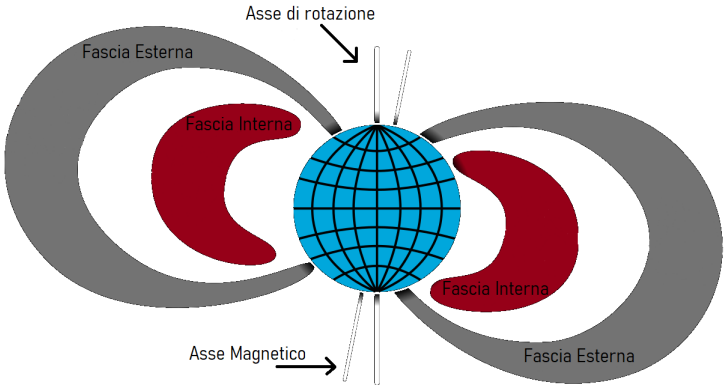

Figura 18: Sezione della fascia di Van Allen.

I raggi cosmici sono costituiti da particelle, come ad esempio elettroni, protoni, ioni pesanti, e da fotoni ad alta energia e di varia origine. Sono conseguenza dell'attività solare, ma possono anche giungere da sorgenti al di fuori del sistema solare o addirittura da altre galassie. L'energia di questo tipo di radiazione è molto elevata e tipicamente si attesta attorno il valore di 10^{20} eV. Il terzo tipo di radiazione, quella di tipo solare, consiste nell'insieme di tutte le particelle emesse dall'attività del Sole. Tra queste è possibile trovare elettroni e protoni la cui energia varia tra i keV e i GeV. [13]

3.2 Effetti delle radiazioni su dispositivi elettronici

Conoscere le caratteristiche e i livelli energetici delle radiazioni che si trovano nell'ambiente spaziale risulta di notevole importanza per caratterizzare gli effetti che queste hanno sui dispositivi elettronici i quali, allo stato attuale, possiedono sempre una certa suscettibilità alle sorgenti ionizzanti. Gli effetti di cui si parla vengono suddivisi in due grandi categorie: **Fenomeni cumulativi** ed **Eventi Singoli**

3.2.1 Fenomeni cumulativi - TID

Con fenomeni cumulativi, spesso indicati con il termine **TID** - Total Ionizing Dose, ci si riferisce all'accumulo di particelle cariche all'interno dell'ossido del terminale di gate dei MOSFET. Quando una qualsiasi radiazione ionizzante colpisce l'ossido di silicio del terminale di gate, l'energia da essa trasportata viene in parte ceduta al materiale ed impiegata per la formazione di una coppia elettrone-lacuna (EHP - Electron-Hole Pair). La densità di EHP generate risulta proporzionale all'energia della radiazione incidente, associata molto spesso all'energia cinetica delle particelle che si possono trovare nell'ambiente spaziale. Con TID (Total Ionizing Dose), si fa riferimento alla quantità totale di energia che deve essere depositata per generare una EHP. La dose di energia necessaria dipende dalle caratteristiche del materiale in considerazione e più precisamente dal gap energetico che sussiste tra la banda di valenza e la banda di conduzione.
Quando le EHP vengono generate, parte di esse vengono subito ricombinate, mentre quelle che rimangono subiscono gli effetti del campo elettrico presente nella regione di ossido. Gli elettroni, a maggiore mobilità, vengono trasportati al di fuori dell'ossido dal campo elettrico; le lacune, a ridotta mobilità, rimangono intrappolate sia all'interno dell'ossido sia all'interfaccia tra ossido e la zona di silicio in cui si forma il canale di conduzione.
L'effetto di tale accumulo porta a conseguenze diverse a seconda che il MOSFET coinvolto sia di tipo p (p-MOSFET) o di tipo n (n-MOSFET), l'effetto globale che però è possibile osservare è una riduzione della mobilità delle cariche di conduzione, fenomeno che quindi degrada le caratteristiche del dispositivo.

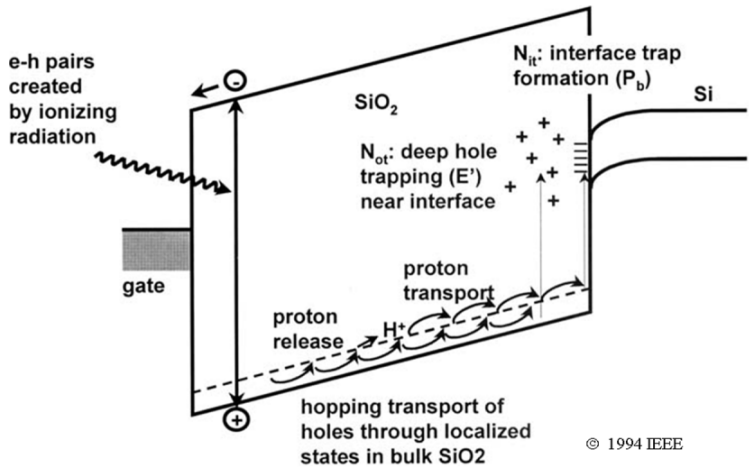

Figura 19: Schema di trasporto delle EHP generate in seguito dell'esposizione a radiazioni. [14]

Particolare attenzione va riposta nelle lacune che vengono intrappolate ai bordi dell'ossido, vicino all'interfaccia con il silicio (interfaccia SiO_2 / Si).
Tali centri di accumulo possono intrappolare o rilasciare le cariche di conduzione dello strato di silicio del canale attraverso l'effetto tunnel. [14]
Come enunciato precedentemente, l'accumulo di tali cariche conduce a conseguenze diverse a seconda del tipo di MOSFET:
n-MOSFET: incremento della corrente di leakage, della corrente di drain a dispositivo acceso e abbassamento della tensione di soglia V_{th}
p-MOSFET: riduzione della corrente di drain a dispositivo acceso, riduzione della corrente di leakage e incremento negativo della tensione soglia V_{th}

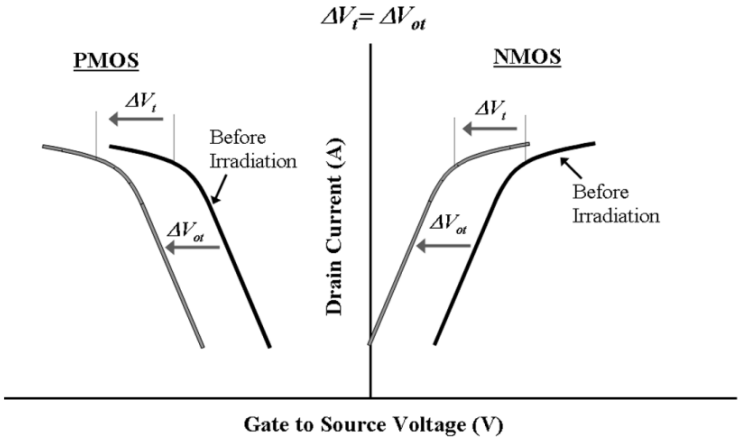

Figura 20: Effetti dell'accumulo di carica su dispositivi MOSFET. [14]

Gli effetti negativi illustrati sono tanto più marcati quanto maggiore è lo spessore dello strato di ossido coinvolto. Per questa ragione, l'avanzare della tecnologia ha consentito una sostanziale riduzione degli spessori degli ossidi di gate, rendendo i dispositivi odierni più resistenti all'esposizione di radiazioni. Il fenomeno però non è stato completamente risolto. Ai margini dei MOSFET esistono degli strati isolanti chiamati STI (Shallow Trench Isolation), che essendo realizzati con spessori di ossido considerevoli costituiscono dei nuovi centri di accumulo di carica e sono quindi all'origine della formazione di correnti di leakage.
L'effetto è osservabile nelle zone di bordo tra tali STI e il silicio del canale di conduzione; la corrente di leakage scorre tra drain e source di uno stesso dispositivo, come visibile in Figura 21.

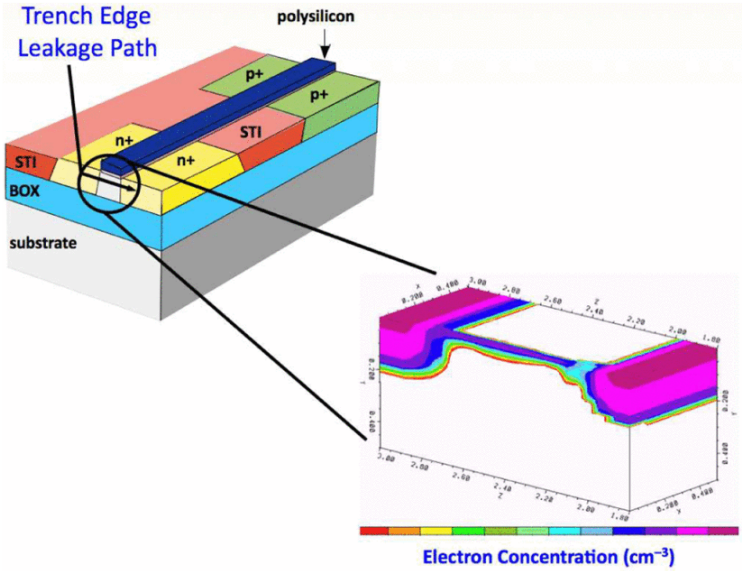

Figura 21: Corrente di leakage tra drain e source nella regione di bordo tra STI e silicio del canale. [15]

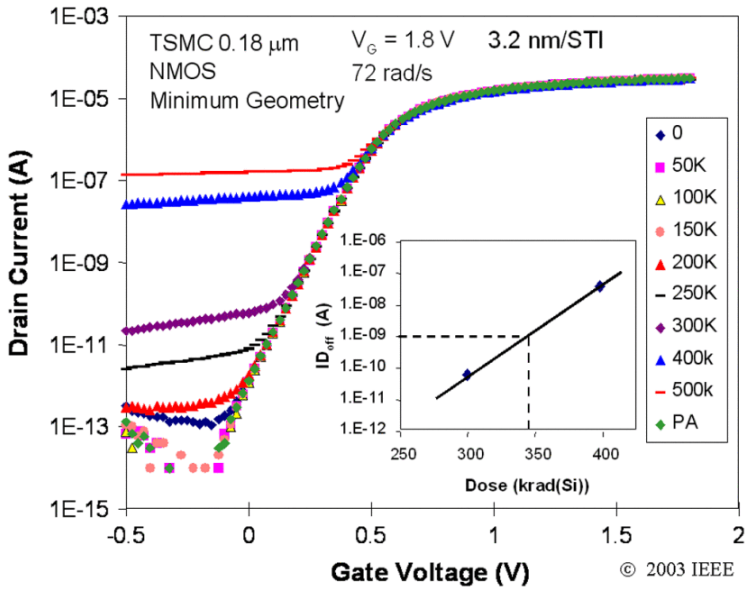

Figura 22: Andamento della corrente di drain in funzione della dose totale. [14]

In Figura 22 è possibile osservare l'andamento della corrente di drain in funzione della dose totale a cui un n-MOS a 180 nm viene esposto. E' interessante osservare come la corrente di leakage a dispositivo spento, cioè per $V_{gs} = 0V$, aumenti in modo pressoché lineare al crescere della dose fino a raggiungere valori dell'ordine dei 100 nA.
Come illustrato precedentemente, l'accumulo delle cariche nell'ossido e nelle STI di un NMOS provoca l'abbassamento della tensione di soglia V_{th} e l'aumento della corrente parassita.
La presenza degli strati isolanti STI in componenti CMOS può generare un fenomeno detto *Interdevice Leakage*, ovvero la generazione di una corrente di leakage tra il drain (o il source) del n-MOSFET e il substrato di tipo n del p-MOSFET. In Figura 23 è possibile osservare il percorso della corrente parassita che si manifesta tra i due MOSFET. Test sperimentali hanno però verificato che la presenza di tale corrente è di gran lunga trascurabile, a parità di dose totale fornita, rispetto alla perdita tra drain e source di uno stesso MOSFET che è stata precedentemente analizzata. [14]

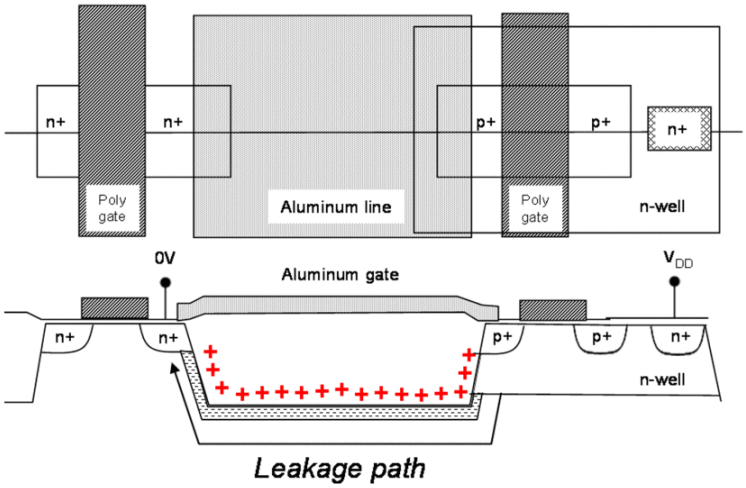

Figura 23: Corrente di leakage tra p-MOSFET e n-MOSFET in tecnologia CMOS. [14]

3.2.2 Eventi singoli - SEE

Con *Eventi singoli*, detti SEE (Single-Event Effect), si intendono gli effetti che si osservano su un dispositivo, come ad esempio una memoria SRAM, un registro, un convertitore AD/DA, in seguito all'esposizione di questo ad una sorgente ionizzante capace di alterarne il funzionamento. Tale alterazione può essere di diversa natura, a seconda della quantità di energia trasferita, ma sopratutto dalla sensibilità del dispositivo in esame.
Gli eventi SEE non sono quindi di tipo cumulativo, ma sono osservabili subito dopo l'esposizione a una sorgente ionizzante.
Si distinguono due tipi di SEE: eventi distruttivi, in conseguenza dei quali il dispositivo risulta danneggiato in modo irreparabile; eventi non distruttivi o transitori, quelli per cui non si verifica un vero e proprio danno fisico, ma i cui effetti potrebbero interferire a livello logico, alterando lo stato di alcune sequenze di bit oppure, nel peggiore dei casi, richiedere il riavvio forzato e/o la riconfigurazione per poter ripristinare le normali condizioni operative. Nelle seguenti sezioni verrà fornita una panoramica di alcuni degli SEE più significativi:

SEE non distruttivi	SEE distruttivi
SEU/MBU	SEL
DSET	
SEFI	

3.2.3 Single Event Upset - SEU

Un Single-Event Upset può essere identificato come quell'evento che provoca il cambio di stato di uno o più bit all'interno di una qualsiasi word di sistema. Volendo identificare una possibile causa di tale fenomeno, è necessario prendere in considerazione gli effetti dello scaling tecnologico. [15] La riduzione delle dimensioni dei MOSFET, con il fine di aumentarne il grado di integrabilità in circuiti complessi, ha condotto ad una diminuzione progressiva della capacità di gate. Poiché tale capacità è associata allo stato del MOSFET, essa costituisce la sede dell'informazione. E' chiaro quindi che la progressiva riduzione della carica che rappresenta l'informazione, costituisce un punto di vulnerabilità attaccabile anche da piccole variazioni causate da dei disturbi esterni, specialmente se questi sono delle radiazioni ionizzanti.
Come si diceva, l'utilizzo dello scaling ha permesso di aumentare il numero di componenti che è possibile realizzare per unità di superficie. L'aumento della densità però, estende il problema del SEU a un numero ancora più elevato di componenti elementari.
Quando gli effetti si estendono a molteplici bit si parla di MBU (Multiple Bit Upset).
Per effettuare una caratterizzazione del fenomeno SEU è possibile eseguire degli specifici test di laboratorio. A tal proposito si riportano alcuni dettagli sui test che possono essere realizzati. In via generale, si utilizza un acceleratore di particelle per indirizzare sul dispositivo sotto test atomi ionizzati, ad esempio rame, cromo, argento, ma anche protoni ed elettroni (Figura 25). Nel test proposto in Figura 24, si vogliono valutare gli effetti di SEU su una memoria SRAM a 65nm quando viene sottoposta al bombardamento di protoni a diversa energia. Come primo passo, si procede a caricare una configurazione di bit nota nella memoria. Successivamente si procede all'irraggiamento, durante il quale viene eseguita una regolare lettura della memoria attraverso l'utilizzo di strumenti esterni. La regolare lettura della memoria è necessaria per evitare di mascherare eventuali SEU multipli che uno stesso bit potrebbe subire. Si termina quindi il test interrompendo l'irraggiamento e passando alla valutazione dei risultati. [16] [15]

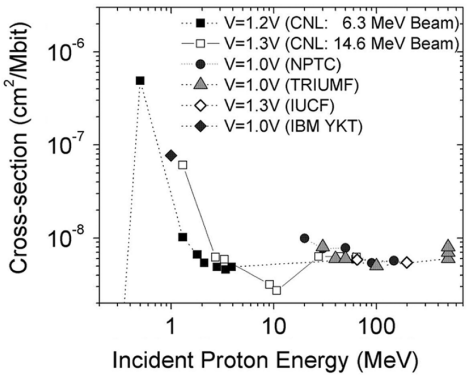

Figura 24: SEU cross-section in funzione dell'energia dei protoni irraggiati. [15]

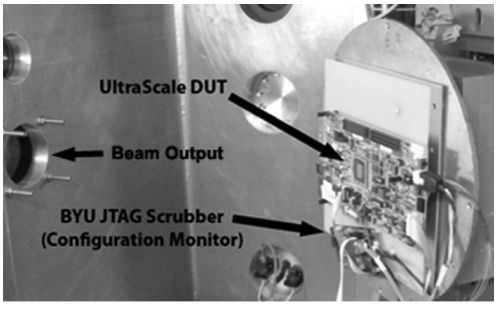

Figura 25: Test setup necessario per condurre l'esperimento. [16]

Nota sull'elaborazione dei dati

Nel grafico di Figura 24 è riportata la cross-section in funzione dell'energia dei protoni incidenti.

Con cross-section ci si riferisce alla probabilità che un SEU si verifichi. Tale probabilità è data da:

$$\sigma_{Mbit}(E) = \frac{\#SEU/Mbit}{\phi_p} \left[\frac{cm^2}{Mbit}\right]$$

1. $\sigma(E)$: cross-section in funzione di E, energia dei protoni incidenti
2. #SEU/Mbit: conteggio degli eventi SEU verificatisi per Mbit. Tale numero viene ottenuto attraverso le regolari letture della memoria e il confronto con la configurazione nota
3. ϕ_p: numero di particelle ionizzanti emesse per unità d'area dall'acceleratore

3.2.4 Digital Single Event Transient - DSET

Con Digital Sigle Event Transient si intende un transitorio in cui una sovratensione (o sovracorrente), indotta dall'esposizione del dispositivo a particelle cariche, si propaga all'interno di un circuito logico. Affinché un tale tipo di transiente generi effettivamente delle conseguenze, è necessario tenere conto del fatto che ciò può avvenire solo se esso si manifesta in una finestra temporale in cui il dispositivo (ad esempio una memoria, un registro o qualsiasi altro circuito logico) è sensibile alle variazioni di tensione (o corrente) al suo ingresso, e cioè quando il transiente risulta allineato con il fronte di salita (o discesa) del segnale di clock o quando risulta avere una durata sufficientemente lunga da poter essere comunque catturato. Se tali condizioni venissero soddisfatte, si potrebbero manifestare dei SEU o anche MBU. [Paragrafo 3.2.3]
I DSET pongono delle serie limitazioni alla frequenza di lavoro del circuito. Si è infatti osservato sperimentalmente che l'aumento della frequenza facilita la propagazione dei transitori. Aumentare la frequenza equivale a richiedere che il dispositivo sia sensibile alle variazioni di tensione ai suoi ingressi per una porzione di tempo sempre più crescente.
In Figura 26 è riportata una simulazione della propagazione di un transiente in una serie di dieci inverter CMOS a 180 nm. Si può osservare come la propagazione tra un inverter e il successivo si manifesti senza particolari attenuazioni e come il transitorio generi un SEU su di un registro di tipo Flip Flop set-reset (RS FF) posto al termine della serie.
Un ulteriore schema esemplificativo di quanto appena descritto viene riportato in Figura 28.

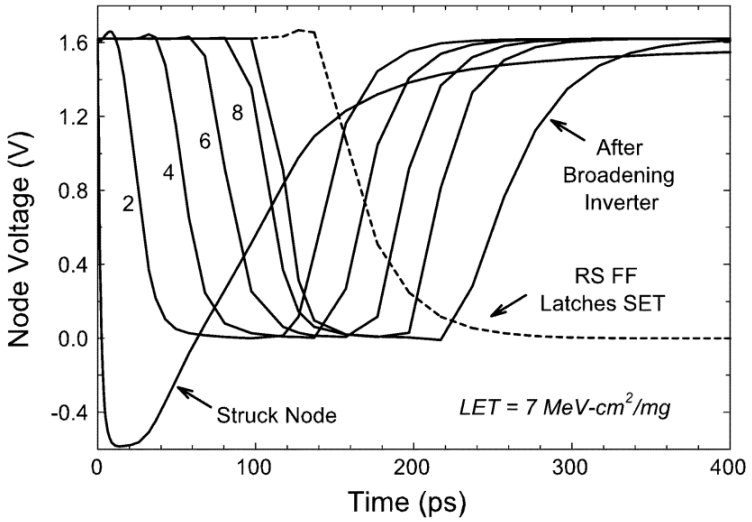

Figura 26: Propagazione di un transiente attraverso una serie di dieci inverter CMOS. [15]

Molteplici esperimenti sono stati condotti per comprendere le proprietà dei DSET con il fine di sviluppare delle tecniche di mitigazione efficaci. In Figura 27 è possibile osservare come lo scaling dei componenti abbia portato ad una maggiore capacità di filtraggio dei transienti e ad una conseguente diminuzione della loro durata.

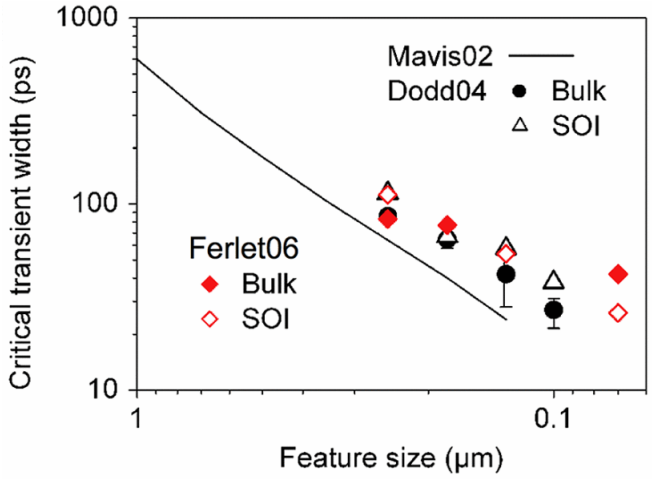

Figura 27: Diminuzione della durata di un transiente in funzione dello scaling del componente. [15]

In conclusione, è possibile individuare quattro criticità che possono tramutare un DSET in un SEU e di cui quindi è necessario tenere conto al fine della mitigazione:

1. La presenza di un percorso logico che conduca ad una memoria o registro;
2. La durata del transiente deve rispettare le necessità del dispositivo, come ad esempio i tempi di set-up e hold in un registro. Maggiore la durata del transiente più difficile la mitigazione;
3. Deve verificarsi in un momento di sensibilità del componente, e cioè essere sincronizzato con i fronti del segnale di clock;
4. La propagazione in assenza di attenuazione può facilitare, specialmente a frequenze elevate, il coinvolgimento di molteplici dispositivi posti sullo stesso percorso logico.

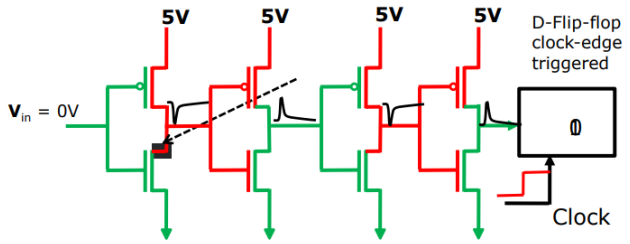

Figura 28: Schema esemplificativo degli effetti di un DSET su una catena di inverter. [13]

3.2.5 Single Event Functional Interrupts - SEFI

I Single Event Functional Interrupts si verificano in seguito ad un SEU o ad un DSET. Le conseguenze però sono di portata notevolmente maggiore. Quando una radiazione ionizzante colpisce un circuito logico con funzioni di controllo, potrebbe causare il totale malfunzionamento del dispositivo controllato. Il recupero delle normali funzionalità potrebbe richiedere il riavvio forzato del dispositivo e in altri casi potrebbe essere necessaria la riconfigurazione dello stesso. A tal proposito, in Figura 29 è riportato l'andamento del numero di errori in una memoria SDRAM bombardata da ioni di Xenon. Inizialmente è possibile osservare il verificarsi di SEU nelle singole celle di memoria; quando il circuito di controllo viene colpito, il numero di errori diventa incontrollabile. Per poter ripristinare il dispositivo è necessario procedere allo spegnimento dello stesso.

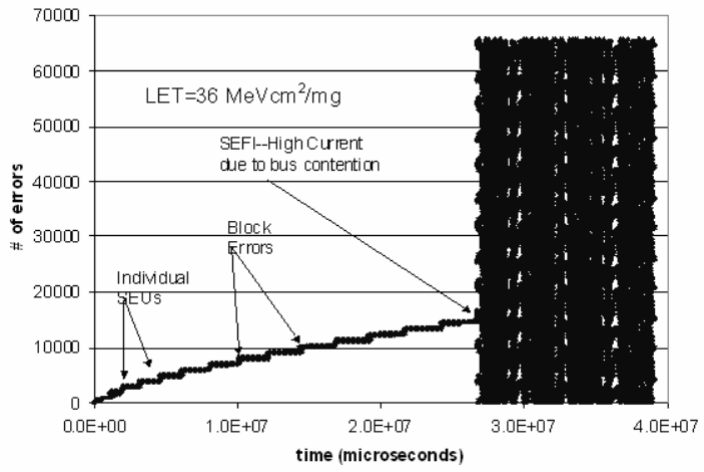

Figura 29: Andamento del numero di errori in una memoria SDRAM. [17]

3.2.6 Single Event Latchup - SEL

Un Single Event Latchup (SEL) può verificarsi in tutti i dispositivi realizzati con tecnologia CMOS. Sebbene al giorno d'oggi il processo produttivo fornisca un certo grado di robustezza, la continua riduzione delle dimensioni dei dispositivi a causa dello scaling comporta l'accrescere della suscettibilità di tali componenti ad eventi SEL.
Intrinseca alla produzione di componenti CMOS è la realizzazione di una regione parassita di tipo pnpn. In Figura 30 è possibile osservare la sezione di un comune inverter CMOS a substrato p in cui la regione pnpn è stata modellata usando i quattro transistor parassiti VPNP 1 & 2 e LNPN 1 & 2.

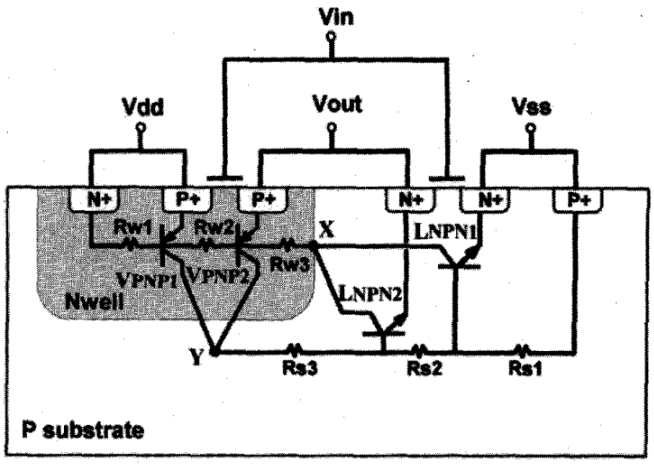

Figura 30: Inverter CMOS e rappresentazione dei componenti parassiti VPNP e LNPN. [18]

Importante notare come sia i transistor verticali (VPNP) che quelli orizzontali (LNPN) abbiano rispettivamente i collettori in comune. Si osserva che parte della corrente di collettore dei VPNP possa raggiungere la base dei transistor LNPN e che la corrente di collettore di questi ultimi contribuisca alla corrente di base dei VPNP. Quello che si è formato rappresenta un loop instabile che, in condizioni normali, risulta interdetto dai cortocircuiti tra base ed emettitore dei rispettivi transistor. In presenza di una sollecitazione esterna però, quando almeno uno di tali transistor viene forzato in conduzione, l'attivazione del loop porta alla formazione di un percorso a bassa impedenza tra i terminali di alimentazione V_{DD} e V_{SS}.

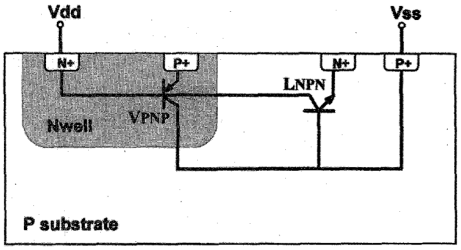

Figura 31: Struttura semplificata della struttura pnpn. [18]

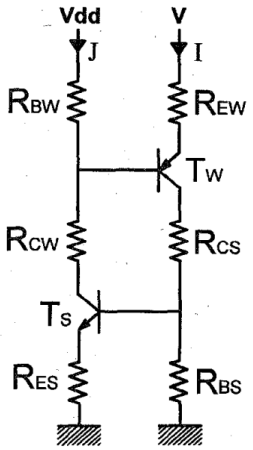

Figura 32: Circuito equivalente della struttura pnpn. [18]

Nelle Figure 31 e 32 viene proposto un modello semplificato che permetterà nel seguito uno studio più agevole dei componenti parassiti coinvolti nella formazione del SEL. Tipicamente, l'arrivo di una radiazione ionizzante proveniente dall'ambiente esterno altera i potenziali all'interno della struttura pnpn, causando l'attivazione degli elementi bipolari parassiti. Si supponga ad esempio che l'impatto della radiazione ionizzante avvenga all'interno della regione Nwell, tra le due isole N+ e P+, e che l'energia da essa depositata sia sufficiente per innescare il fenomeno del latchup.
Il transistor PNP verticale entra in conduzione a causa dell'alterazione del potenziale all'interno della regione. Le lacune della corrente di collettore vengono iniettate nel substrato P, in cui diffondono. La corrente ad esse associata risulta sufficiente per portare in conduzione anche il transistor NPN orizzontale. Quando questo avviene, allora la corrente di collettore dell'NPN genera un flusso di elettroni verso la regione Nwell, di fatto alimentando la base del PNP. Tale sequenza di eventi rappresenta il loop instabile prima descritto. Con i transistor parassiti in conduzione, i terminali di alimentazione sono connessi da un percorso a bassa impedenza, favorevole al passaggio di correnti elevate perché prossimo ad un cortocircuito. Ciò potrebbe causare la rottura del componente per effetto termico. Nel caso in cui il componente risultasse ancora funzionante dopo un SEL, ciò non potrebbe escludere il manifestarsi di serie limitazioni di affidabilità dello stesso. Una tale condizione è estremamente difficile da rilevare e potrebbe compromettere l'intera operazione in cui il dispositivo è impiegato. [18]

Per mitigare gli effetti di un SEL la tecnica privilegiata è quella di ridurre al minimo la possibilità di interazione tra le strutture parassite prima descritte. In Figura 33 è possibile osservare che suddividendo il substrato, normalmente drogato p, in due componenti a diversa concentrazione di drogante p^+ e p^-, si possano indirizzare le lacune verso la profondità del substrato, rendendo più difficoltosa l'accensione del transistor LNPN. Altre tecniche prevedono invece l'isolamento del PMOS dall'NMOS attraverso l'utilizzo di STI, ovvero strati di ossido isolante la cui applicazione risulta però associata ad ulteriori considerazioni già analizzate al paragrafo [3.2.1].

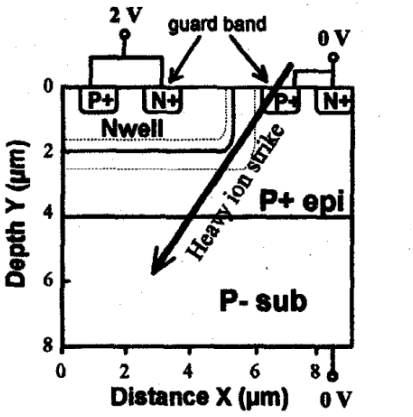

Figura 33: Tecnica di mitigazione dei SEL. [18]

3.3 Tolleranza alle radiazioni per FPGA Kintex UltraScale

Dopo aver esaminato le caratteristiche dei principali effetti a cui qualsiasi dispositivo elettronico è esposto quando impiegato nell'ambiente spaziale, in Tabella 4 si riportano i risultati sperimentali ottenuti per la famiglia di FPGA Kintex UltraScale.
In particolare, è interessante osservare come i dati relativi gli eventi SEU vengano riportati sia per la memoria di configurazione (CRAM), sia per la memoria Block RAM (BRAM) nell'ipotesi di utilizzo in orbite GEO (Orbite Equatoriali Geosincrone). Tali risultati sono stati ottenuti attraverso l'utilizzo di modelli matematici capaci di descrivere il verificarsi di SEE in un certo ambiente spaziale. In questo caso il modello utilizzato è il CREME96. [19]

Fenomeno	**Minimo**	**Tipico**	**Massimo**	**Unità**
TID (GEO)	-	100	120	Krad (Si)
SEL	-	80	-	$MeV\text{-}cm^2/mg$
SEU_{CRAM} (GEO)	-	$9.5 \cdot 10^{-9}$	-	Upset/bit/giorno
SEU_{BRAM} (GEO)	-	$2.3 \cdot 10^{-8}$	-	Upset/bit/giorno
SEFI_{CRAM} (GEO)	-	$4.5 \cdot 10^{-4}$	-	Upset/device/giorno

Tabella 4: Risultati sperimentali dell'esposizione alle radiazioni per FPGA Kintex. [19]

In generale, i risultati sperimentali come quelli forniti in tabella, vengono utilizzati al fine della progettazione di opportuni sistemi di mitigazione per contrastare gli effetti negativi associati alle radiazioni. L'adeguatezza di un componente dipende dalle condizioni in cui esso dovrà operare. Questo implica che prima della scelta dei componenti si renda necessaria la caratterizzazione, attraverso un modello matematico adatto, dell'ambiente spaziale di interesse. Definite le caratteristiche di quest'ultimo, si procederà alla scelta dei componenti compatibilmente alle necessità operative. La risposta dei componenti alle radiazioni viene ricavata da risultati sperimentali forniti direttamente dal venditore o ricavati personalmente eseguendo delle prove di laboratorio. Poiché condurre test per tutte le possibili condizioni operative risulterebbe estremamente costoso in termini economici e temporali, la progettazione della mitigazione si pone come obiettivo la gestione del caso peggiore in cui tale dispositivo potrebbe ricadere. Questo però introduce ulteriori problematiche. Le assunzioni fatte per determinare il caso peggiore potrebbero rivelarsi errate, le condizioni di test potrebbero non essere rappresentative delle condizioni operative reali o ancora, il componente testato potrebbe essere diverso, in termini di risposta alle radiazioni, da quello effettivamente impiegato per la missione.

Dal datasheet di Figura 4 è possibile ottenere informazioni sulla Total Ionizing Dose (TID), la massima dose di radiazioni ionizzanti a cui il dispositivo Kintex UltraScale può essere esposto. Tale dato può essere contestualizzato sfruttando il modello SPENVIS per calcolare la dose di radiazione ionizzante presente in un'orbita GEO. Ai fini della simulazione, si può pensare di inserire il dispositivo di cui si vuole quantificare l'esposizione alle radiazioni al centro di una sfera (Figura 34). Quest'ultima si suppone essere di alluminio e rappresenta una sorta di scudo, o *shield*, che potrebbe essere utilizzato per proteggere il dispositivo una volta installato, ad esempio, all'interno di un satellite.
Il modello SPENVIS permette di calcolare la dose di radiazione ionizzante a cui il dispositivo è esposto al centro della sfera simulando un'orbita GEO.

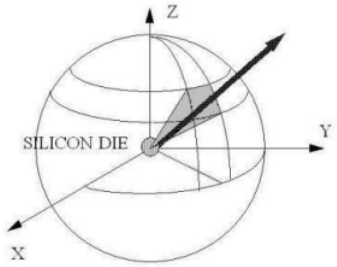

Figura 34: Posizionamento del dispositivo al centro della sfera per il calcolo della dose di radiazione ionizzante. [13]

In Figura 35 sono riportate le dosi al centro della sfera al variare dello spessore dello shield e per un periodo di esposizione pari a 18 anni, simulando l'ambiente spaziale corrispondente a un'orbita GEO. Si può apprezzare come l'aumento dello spessore comporti una diminuzione della dose totale al centro della sfera. [13]
Combinando i dati ottenuti dal datasheet e dal modello, è possibile affermare che, usando uno shield di alluminio con spessore pari a circa 6 mm, sarebbe possibile utilizzare un FPGA Kintex UltraScale per 18 anni in un'orbita GEO.

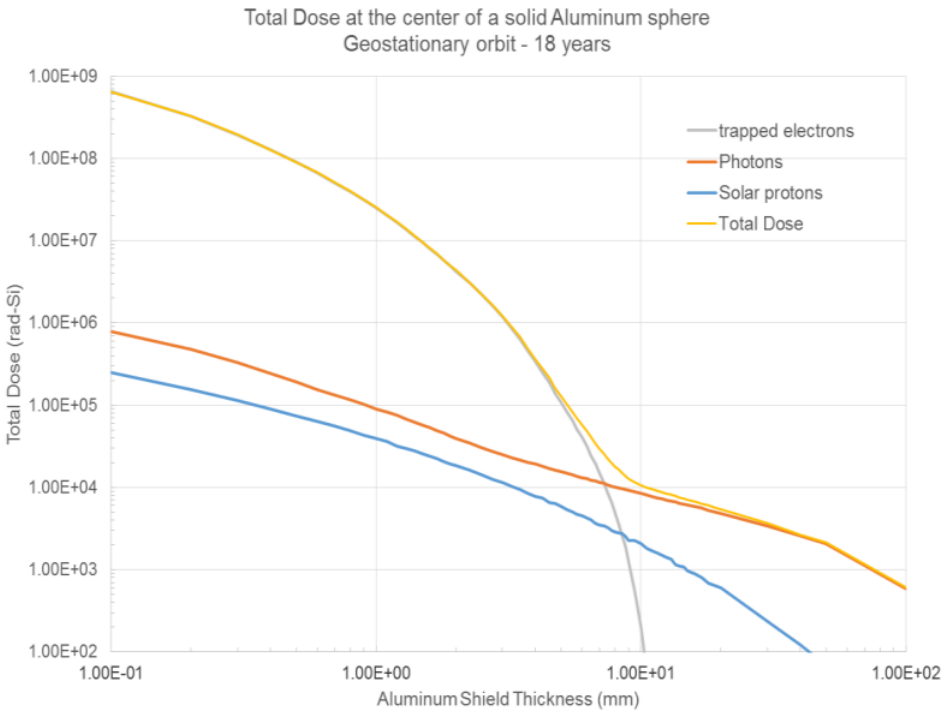

Figura 35: In arancione, la Total Ionizing Dose al centro di una sfera d'alluminio in un'orbita GEO per un'esposizione di 18 anni in funzione dello spessore dello shield. [13]

3.3.1 Nota sulle unità di misura: rad & LET

Nei grafici fino ad ora riportati si fa molto spesso riferimento a particolari unità di misura, utilizzate per specificare l'interazione tra radiazioni ionizzanti e materia.
Si ritiene quindi opportuno specificare la natura di tali unità.

- *rad*: utilizzato per descrivere la quantità di energia depositata da una radiazione ionizzante in un certo materiale per unità di massa. In molti grafici il materiale considerato è il silicio (Si).
  Vale la seguente relazione: $1\ rad = 0.01\ \frac{J}{Kg}$

- *LET* - Linear Energy Transfer: inteso come il rapporto tra l'energia che una particella rilascia per unità di lunghezza in un certo materiale e la densità di quest'ultimo, in questo caso supposto essere silicio (Si).
  In formule: $\frac{MeV}{cm} \div \frac{mg}{cm^3} = \frac{MeV \cdot cm^2}{mg}$

4 Tecniche di mitigazione

4.1 Classificazione dei bit sensibili

Come annunciato nei paragrafi precedenti, gli FPGA sono dispositivi suscettibili all'esposizione delle radiazioni tipiche dell'ambiente spaziale in cui possono essere impiegati. Per raggiungere l'elevato grado di affidabilità richiesto è necessario implementare delle strategie di mitigazione efficaci per cercare di ridurre al minimo gli effetti negativi delle radiazioni. Allo stesso tempo è necessario limitare i costi, sia in termini di risorse utilizzate, sia dal punto di vista della complessità realizzativa dei sistemi di rilevazione e correzione degli errori.
Nei seguenti paragrafi saranno analizzate alcune strategie che comunemente sono implementabili a bordo di FPGA, inclusi quelli della famiglia Kintex UltraScale, e che consentono di ottenere dei risultati particolarmente soddisfacenti.
Come annunciato al paragrafo 2.5, i dati riguardanti la configurazione dell'FPGA sono conservati all'interno della memoria di configurazione. Per dati di configurazione si intendono sequenze di bit che vengono utilizzate per programmare le LUT dei CLB, per gestire le interconnessioni e tutti gli altri dispositivi presenti a bordo. Tale memoria però, come osservato al paragrafo 3.2.3, può subire eventi di tipo SEU, ovvero bit-flip.
Il cambiamento dei bit della memoria di configurazione è particolarmente grave, in quanto ciò corrisponde ad alterare il circuito predisposto dal programmatore, interferendo quindi con l'implementazione delle diverse funzioni logiche e delle interconnessioni tra elementi base di un FPGA. Per comprendere al meglio tale problema è utile condurre una classificazione dei bit di configurazione in base alle conseguenze che si verificano in seguito ad un loro cambiamento di stato. La memoria di configurazione è suddivisa in blocchi elementari chiamati frame. Ciascun frame è necessario per effettuare la programmazione di una certa sotto-sezione della matrice dell'FPGA, permettendo la configurazione di un certo numero di CLB, di interconnessioni e di blocchi di I/O. Attualmente, negli FPGA UltraScale di Xilinx, un frame corrisponde a 3936 bit. Un bit di configurazione è detto *sensibile* se il suo cambiamento di stato apporta delle modifiche al funzionamento del circuito implementato. In presenza di un bit sensibile, il malfunzionamento può essere di due tipi:

1. Non persistente: il cambiamento di stato del bit sensibile provoca il malfunzionamento del circuito. Se il bit coinvolto viene riportato allo stato corretto, allora il circuito torna alle normali condizioni operative e non sono richieste ulteriori azioni.

2. Persistente: il cambiamento di stato del bit sensibile provoca il malfunzionamento del circuito. Anche se il bit viene riparato, il circuito coinvolto non ritorna nelle normali condizioni operative.

I bit classificati come persistenti generano quindi gravi malfunzionamenti del circuito nel caso in cui essi subiscano un SEU; di fatto essi sono all'origine degli eventi SEFI. [20]
In Figura 36 è possibile osservare il comportamento delle due tipologie di sensibilità.
Tali grafici sono stati realizzati confrontando i segnali di uscita di due dispositivi: uno di riferimento e uno esposto a radiazioni. Quando il bit sensibile viene modificato è possibile osservare una differenza tra i segnali di uscita dei due dispositivi.

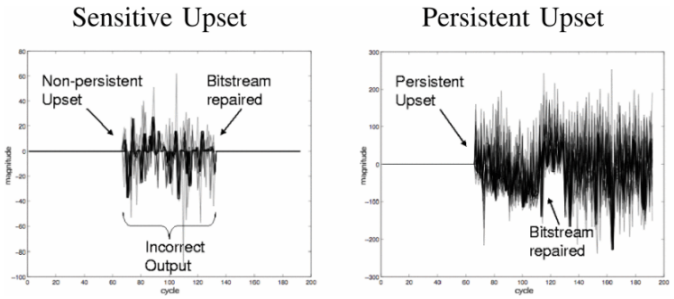

Figura 36: Andamento della differenza dei segnali di uscita tra un dispositivo di riferimento e uno sottoposto a radiazioni. A sinistra gli effetti del cambiamento di stato di un bit non persistente, a destra quelli di un bit persistente. [20]

I circuiti che sono coinvolti in una catena di feedback sono quelli maggiormente associati a bit di configurazione persistenti. Il motivo di ciò risiede nel fatto che se il circuito varia in seguito al cambiamento dei bit di configurazione, i segnali di uscita che vengono prodotti, propagati dalla struttura di feedback e utilizzati come ingressi per l'elaborazione successiva, potrebbero corrompere il normale funzionamento del circuito in un modo del tutto imprevedibile anche nel caso in cui i bit coinvolti venissero riparati.
Riconoscere le strutture circuitali che potrebbero generare dei bit di configurazione persistenti è quindi di fondamentale importanza. [20]

A tal proposito, in [20], viene proposto un metodo per poter classificare i bit di configurazione secondo quanto appena descritto. E' necessario disporre di due FPGA, di cui uno svolge il ruolo di riferimento, *Golden Device*, mentre l'altro è il *DUT - Device Under Test*. Un dispositivo esterno accede alla memoria di configurazione del DUT utilizzando una delle interfacce disponibili (par. 2.5) e procede a modificare i bit di configurazione uno alla volta, simulando il verificarsi di un SEU. Contemporaneamente, le uscite dei due dispositivi vengono confrontate e se queste differiscono allora il bit viene classificato come *sensibile*. Successivamente, il bit viene riportato allo stato originario e le uscite vengono nuovamente confrontate. Se queste sono ancora discordanti, allora il bit è anche persistente. Ripetendo tale operazione per tutti i bit contenuti nella memoria di configurazione è possibile ricavare il profilo di sensibilità dell'intera memoria CRAM.
In Figura 37 è possibile osservare tre immagini:

- Figura 37.(a) rappresenta le risorse dell'FPGA utilizzate per la realizzazione di un kernel DSP;
- Figura 37.(b) è la distribuzione dei bit sensibili all'interno della memoria di configurazione, ricavati con il metodo prima enunciato;
- Figura 37.(c) è la porzione di bit sensibili che sono anche persistenti.

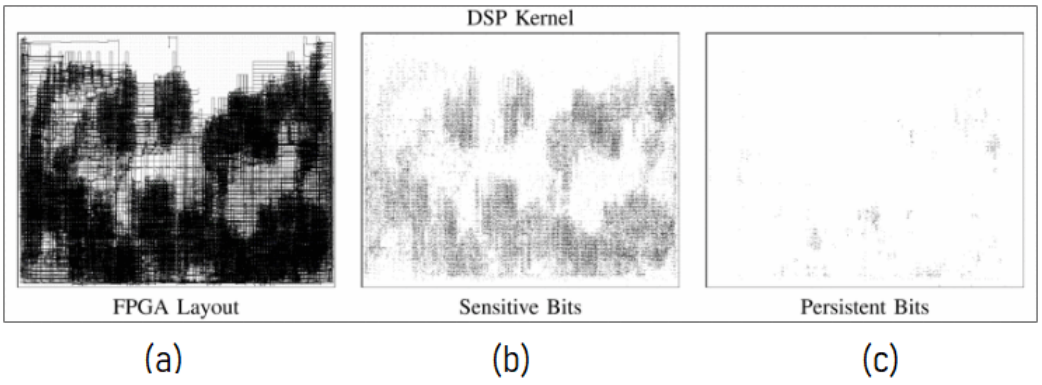

Figura 37: Realizzazione di un kernel DSP: utilizzo delle risorse dell'FPGA; distribuzione dei bit di configurazione sensibili e persistenti. [20]

4.2 Triple Modular Redundancy - TMR

Con Triple Modular Redundancy si intende la triplicazione di una parte dei circuiti che vengono implementati a bordo dell'FPGA e l'utilizzo di un sistema di votazione. Il sistema di voting ha la funzione di confrontare i segnali di uscita delle tre copie per scegliere quale output sia corretto secondo un criterio di maggioranza. Seguendo la classificazione introdotta al paragrafo precedente, i circuiti che per primi devono essere triplicati sono quelli associati a dei bit di configurazione persistenti. Questa scelta è dettata dalla limitata disponibilità di risorse, non sarebbe infatti possibile triplicare tutti i circuiti implementati dal programmatore.
La definizione del circuito di voting è a sua volta conservata all'interno della memoria di configurazione. Ciò significa che anche tale sistema è suscettibile agli eventi SEU prima descritti. Nel caso in cui sia presente un solo voter, questo costituisce un punto di debolezza non indifferente. In alcuni casi particolari è possibile implementare la triplicazione del voter, rendendo però necessarie ulteriori strutture di controllo e decisione che non verranno considerate in questa discussione.
L'implementazione del TMR inizia con l'individuazione dei circuiti considerati suscettibili, quelli per cui un anche un singolo SEU potrebbe generare un SEFI. Una volta fatto ciò, si sceglie una strategia di triplicazione, di cui vengono forniti degli esempi in Figura 38.

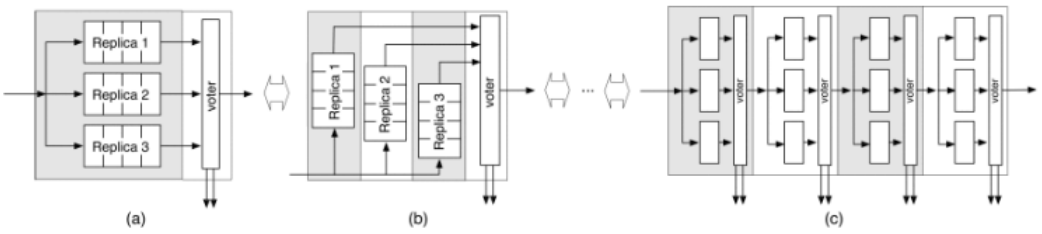

Figura 38: Possibili strategie di TMR. [21]

- Figura 38.(a) La configurazione delle tre copie del circuito è contenuta all'interno dello stesso frame. La configurazione del circuito di voting è contenuta in un frame diverso dal precedente.
- Figura 38.(b) Ogni copia del circuito è configurata in frame distinti. La configurazione del circuito di voting è contenuta in un frame diverso dai precedenti.
- Figura 38.(c) La configurazione delle copie del circuito e del sistema di voting sono contenute all'interno di uno stesso ed unico frame.

In generale, l'implementazione manuale del TMR risulta particolarmente complicata. Per di più, gli strumenti di sviluppo che traducono il sorgente VHDL nel flusso di bit di configurazione per l'FPGA potrebbero interferire con tali moduli, rimuovendoli in quanto apparentemente repliche dello stesso circuito. Molti ambienti di sviluppo non sono "TMR-aware", cioè non riconoscono queste strutture circuitali come
implementazioni della tecnica TMR e per questo, nel tentativo di ridurre al minimo

l'occupazione d'area e l'utilizzo delle risorse, rendono del tutto inefficace tale strategia di mitigazione. [22]
Per sopperire a tale comportamento, sono stati sviluppati degli strumenti appositi in grado di gestire al meglio le realizzazioni TMR.
A bordo degli FPGA della famiglia Kintex viene messo a disposizione l'utilizzo del modulo *MicroBlaze TMR*, una *IP* da integrare all'interno dell'ambiente di sviluppo Vivado. Il MicroBlaze è un soft-processor RISC (Reduced Instruction Set Computer) a 32-bit ottimizzato per eseguire applicazioni embedded. Essendo un soft-processor, esso viene implementato direttamente a bordo dell'FPGA utilizzando le risorse logiche di quest'ultimo.
In Figura 39 si osserva lo schema a blocchi del processore MicroBlaze, in cui è possibile riconoscere gli elementi tipici di un microprocessore: il program counter, il decoder per le istruzioni, i registri, l'unità logico-aritmetica (ALU) e l'unità di controllo della memoria. Per l'accesso alla memoria il processore MicroBlaze utilizza il bus LMB (Local Memory Bus) avente linee separate per istruzioni e dati. [23]

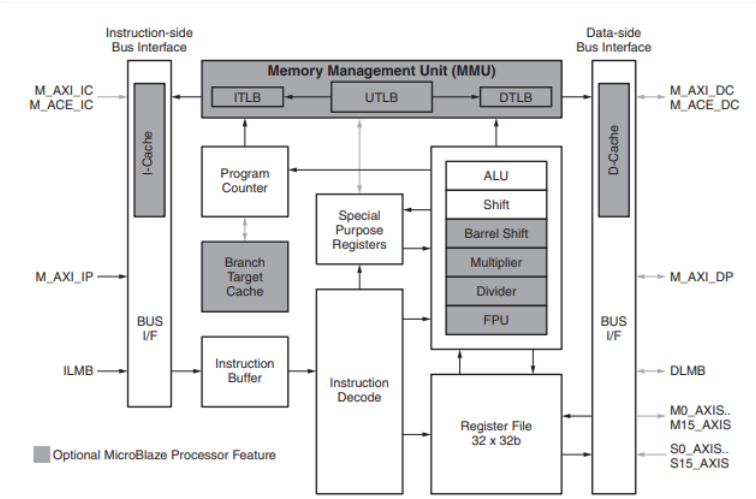

Figura 39: Schema a blocchi del processore MicroBlaze. [23]

In Figura 40 è invece riportato il processore MicroBlaze sotto forma di modulo IP elementare.

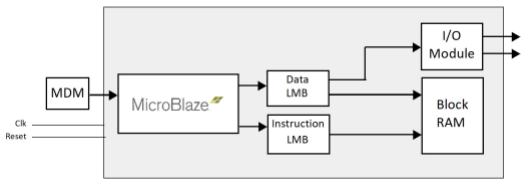

Figura 40: Modulo IP MicroBlaze elementare. [24]

Il *MicroBlaze TMR* implementa la tecnica del TMR triplicando il modulo base MicroBlaze e aggiungendo i circuiti di votazione necessari. Sono quindi disponibili tre processori separati, ma configurati con le stesse funzioni logiche e regolati da un sistema di voting.
Il programmatore ha la possibilità di scegliere diverse configurazioni. Per incrementare la robustezza del sistema, la memoria Block RAM (BRAM) può essere triplicata.
Gli accessi in lettura e scrittura vengono regolati attraverso l'uso di un ulteriore voter per garantire che tutti e tre i processori facciano riferimento alle stesse aree di memoria durante l'elaborazione. [Figura 41]
In alternativa, per ridurre l'utilizzo delle risorse, è possibile utilizzare una singola memoria BRAM. In quest'ultima opzione, la maggiore vulnerabilità del sistema viene mitigata proteggendo i dati in memoria attraverso l'utilizzo di codici ECC. [Figura 42] [24]

Nelle Figure 41 e 42 è possibile osservare le possibili configurazioni MicroBlaze appena discusse.

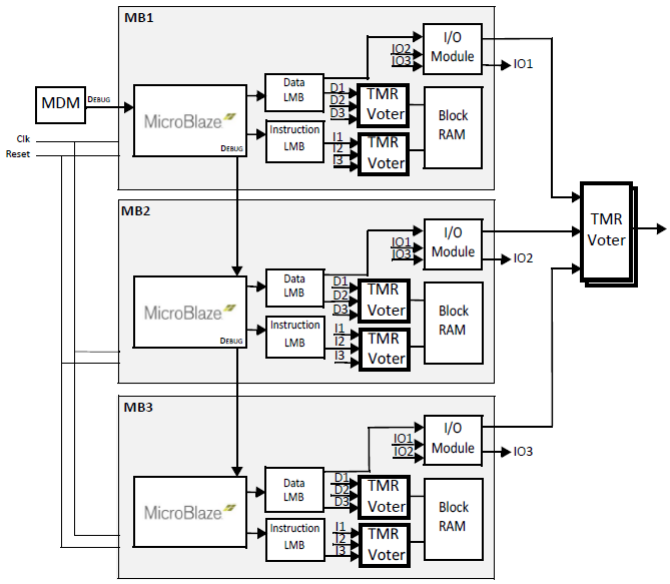

Figura 41: Configurazione MicroBlaze con BRAM triplicata. [24]

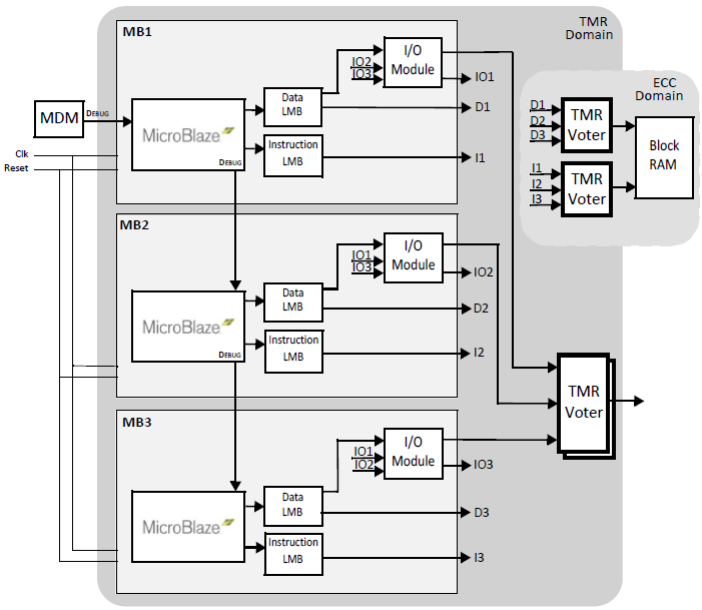

Figura 42: Configurazione MicroBlaze a singola BRAM protetta da ECC. [24]

4.3 Codici EDAC

Come si è visto, lo stesso circuito di voting impiegato in una tecnica TMR potrebbe essere modificato in seguito ad eventi SEU sui bit di configurazione ad esso associati.
Per tale ragione, si rende necessario l'utilizzo combinato di diverse strategie di rilevazione e correzione degli errori.
A bordo degli FPGA UltraScale di Xilinx è possibile:

- Usare codici CRC per rilevare la presenza di errori in un flusso di bit.
- Usare codici ECC per correggere errori.

Prima di procedere con la descrizione dell'implementazione a bordo degli FPGA, si riportano le caratteristiche principali dei codici sopra citati.

4.3.1 Codici CRC

I Cyclic Redundancy Check (CRC) sono codici che vengono utilizzati per rilevare, ma non correggere, la presenza di errori in un certo messaggio. Tale tecnica ha lo scopo di generare una sequenza di bit di controllo in funzione del messaggio di interesse. Il messaggio e la sequenza di controllo formano poi un'unica parola di codice. Quando il destinatario riceve la parola di codice, è possibile verificarne l'integrità procedendo nuovamente al calcolo della sequenza di controllo utilizzando il messaggio ricevuto. Se la sequenza ottenuta e quella presente nel codice coincidono, allora il messaggio è privo di errori.

4.3.2 Codici ECC

Con Error Correction Code (ECC), o codice di Hamming, si fa riferimento alla tecnica per cui k bit di informazione vengono codificati in una parola di codice di n bit, aggiungendo n-k bit di controllo attraverso l'utilizzo di un codificatore dedicato. Il processo di verifica richiede invece un decodificatore, capace di non solo rilevare la presenza di errori, ma anche di correggerli.
Tipicamente i codici ECC più utilizzati sono quelli di tipo SECDED, Single-Error Correction and Double-Error Detection, che consentono di rilevare due errori e di correggerne al massimo uno.

4.3.3 Global CRC

Un intero bitstream di configurazione è associato a 32 bit di controllo CRC. Questi bit sono utilizzati per validare l'intero contenuto della memoria di configurazione.
Il calcolo del CRC e il confronto con la sequenza di controllo fornita richiede un tempo considerevole. Prima che un errore possa essere rilevato, è infatti necessario che l'intero bitstream sia caricato in memoria. Nonostante questo, il meccanismo di CRC risulta estremamente robusto ed efficace.

4.3.4 Frame ECC

Ciascun frame della memoria di configurazione viene codificato da un codice ECC con sequenza di controllo a 32 bit. Questo permette di rilevare e correggere gli errori che si verificano al livello di un singolo frame. Come detto precedentemente, poiché i codici ECC maggiormente usati sono del tipo SECDED, è possibile che errori multipli non vengano rilevati al livello di frame, ma sicuramente lo saranno attraverso il Global CRC.

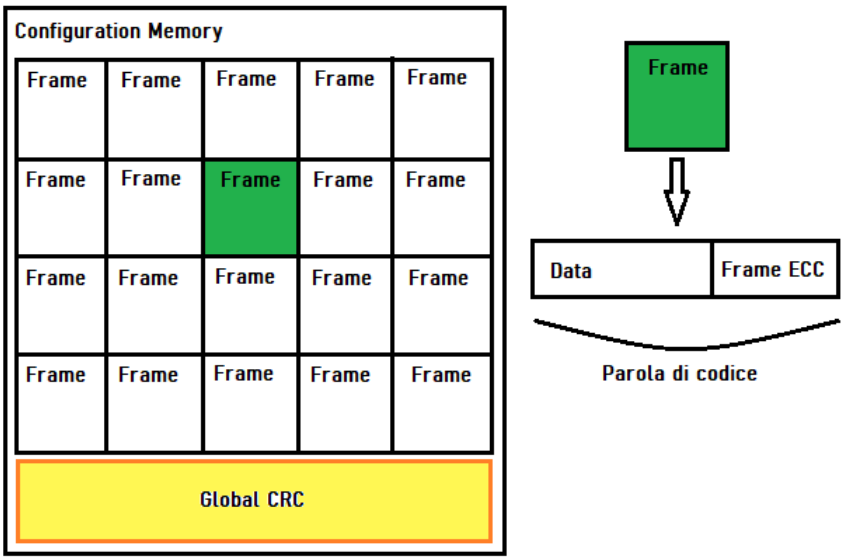

Figura 43: Rappresentazione schematica della memoria di configurazione.

4.4 Configuration Scrubbing

Con il termine Configuration Scrubbing si intende l'accesso periodico alla memoria di configurazione da parte di un sistema dedicato al controllo della stessa e senza interferire con il normale funzionamento dell'FPGA. Fino ad ora, si è discusso di come sia possibile codificare i bit di configurazione utilizzando dei codici ECC. Risulta però necessario disporre di un circuito dedicato per prevenire l'accumulo di bit flip multipli e per mantenere intatta la configurazione predisposta dal programmatore. Lo scrubber, ovvero il circuito che realizza tale operazione, accede sequenzialmente ai frame della memoria di configurazione. Per ognuno di essi viene eseguita le decodifica del codice ECC. Se dopo l'elaborazione vengono rilevati degli errori, allora esso procede alla riparazione degli stessi. I circuiti di scrubbing possono essere realizzati sia internamente che esternamente all'FPGA e configurati per soddisfare diverse richieste operative. Di seguito sono riportati alcuni esempi delle configurazioni maggiormente utilizzate. [11]

- **Rilevazione e correzione tramite ECC**
  Questo tipo di scrubber elabora il codice ECC del frame di configurazione per rilevare la presenza di errori. Se questi sono presenti allora viene sfruttata la capacità correttiva dei codici di Hamming che nel caso degli SECDED permette di correggere fino a un bit-flip.

- **Rilevazione tramite ECC e correzione tramite Golden Memory**
  In questo caso, il codice ECC viene decodificato ed utilizzato solo per rilevare la presenza di errori. La correzione prevede la riscrittura del singolo frame danneggiato utilizzando la sua copia corrispondente custodita in una memoria apposita, la "Golden Memory", attraverso il processo di riconfigurazione parziale precedentemente descritto.

Nella Figura 44 è possibile osservare lo schema di principio per il collegamento di uno scrubber all'FPGA e ad una Golden Memory.

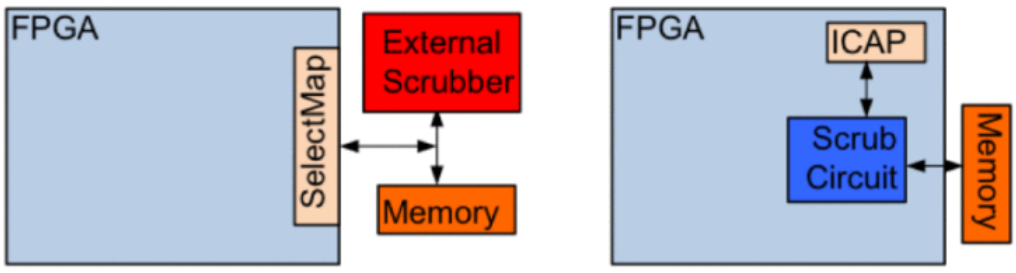

Figura 44: Esempi di Configuration Scrubber con Golden Memory. [25]

Un altro importante aspetto riguardante i Configuration Scrubber è rappresentato dall'interfaccia utilizzata per l'accesso alla memoria. Le interfacce maggiormente utilizzate sono state riportate al paragrafo 2.5. [25][11]
La scelta di implementare scrubber interni piuttosto che esterni porta con sé una serie di aspetti sia positivi che negativi. Per implementare scrubber esterni è richiesto l'utilizzo di un dispositivo appositamente dedicato, ad esempio un microprocessore oppure un ulteriore FPGA, comportando un incremento del consumo di potenza del sistema.
Del resto, sebbene gli scrubber interni non richiedano l'utilizzo di ulteriori componenti esterni, è bene osservare che la loro implementazione richieda una porzione della logica dell'FPGA e che quindi il quantitativo di risorse disponibili al programmatore si riduca.
Infine, gli scrubber interni sono esposti agli eventi SEU della memoria di configurazione e quindi è possibile che i bit di configurazione associati al circuito di scrubbing vengano corrotti.
Al fine di facilitare lo sviluppo di tutti gli accorgimenti necessari alla mitigazione degli eventi SEU ai danni della memoria di configurazione, Xilinx ha sviluppato il modulo Soft Error Mitigation IP (SEM IP). Integrando tale IP all'interno dell'ambiente di sviluppo Vivado, è possibile usufruire di meccanismi di scrubbing già pronti e testati da Xilinx stessa. Il sistema SEM IP sfrutta il Frame-ECC e il Global-CRC per rilevare e correggere gli errori. Per accedere alla memoria di configurazione il sistema utilizza l'interfaccia ICAP. Il circuito viene implementato sullo stesso FPGA su cui viene operato lo scrubbing e proprio per questo motivo è necessario ricordare come anche tale sistema sia vulnerabile agli eventi SEU della memoria di configurazione. [26]
In Figura 45 è possibile osservare il modulo SEM IP e i collegamenti richiesti per il suo corretto funzionamento.

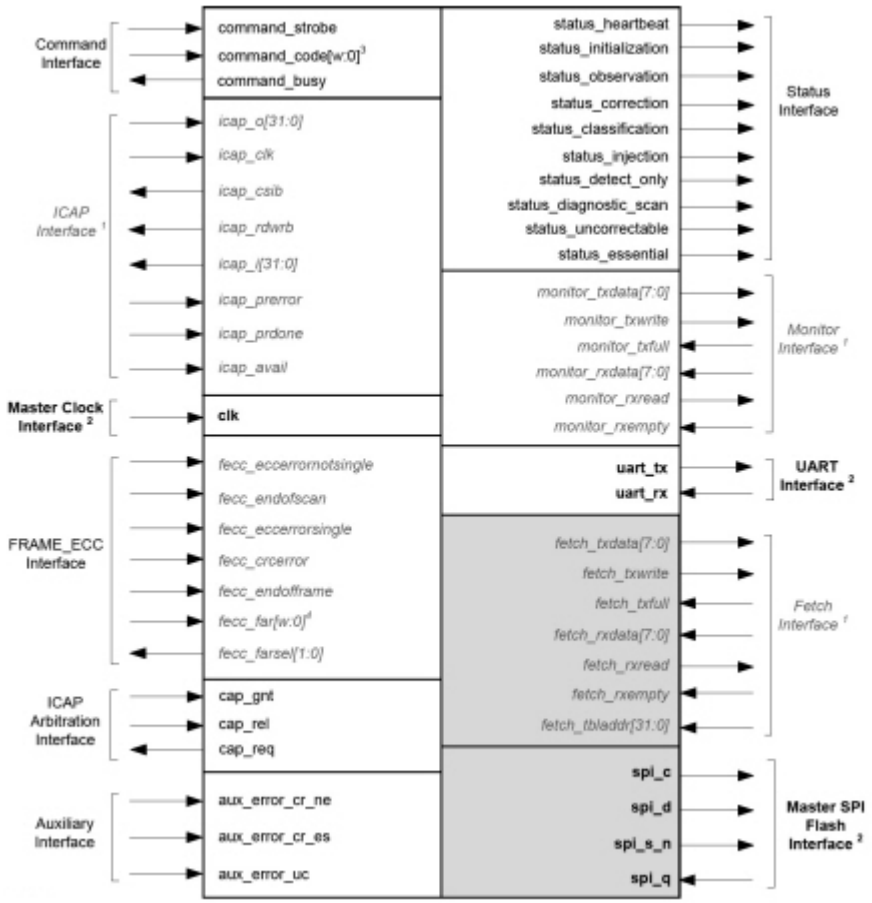

Figura 45: Schema dei collegamenti del sistema SEM IP. [26]

Qui di seguito vengono brevemente descritte le funzioni svolte da alcune delle interfacce che tale IP mette a disposizione del programmatore: [26]

- **ICAP Interface**: permette la connessione tra il Controller SEM e l'interfaccia ICAP usata per accedere alla memoria di configurazione;
- **FRAME_ECC Interface**: permette al Controller SEM di utilizzare le funzioni del FRAME_ECC per rilevare gli errori all'interno della memoria di configurazione;
- **Status Interface**: permette di analizzare lo stato del Controller SEM;
- **Command Interface**: consente di forzare degli errori nei bit contenuti nella memoria di configurazione. In fase di test, ciò è particolarmente utile perché permette di simulare il verificarsi di eventi SEU, ed è quindi possibile verificare la risposta del sistema SEM e il suo corretto funzionamento;
- **Monitor Interface**: usata per interfacciarsi con il Controller SEM, ingloba le funzioni delle precedenti due interfacce;
- **Fetch Interface**: consente al Controller SEM di richiedere dati da una sorgente esterna. Tale operazione potrebbe rendersi necessaria durante la fase di correzione degli errori nella memoria di configurazione.

5 Conclusioni

5.1 Esempio Applicativo: SpaceCube v3.0 Mini

Con SpaceCube ci si riferisce alla piattaforma hardware sviluppata dal Goddard Space Flight Center (GSFC) della NASA con l'obiettivo di creare un sistema modulare che potesse essere facilmente integrato all'interno di nanosatelliti e che avesse un costo realizzativo ridotto, ma con elevate capacità di calcolo. In Figura 46 viene riportato lo schema a blocchi della versione SpaceCube v3.0 Mini.

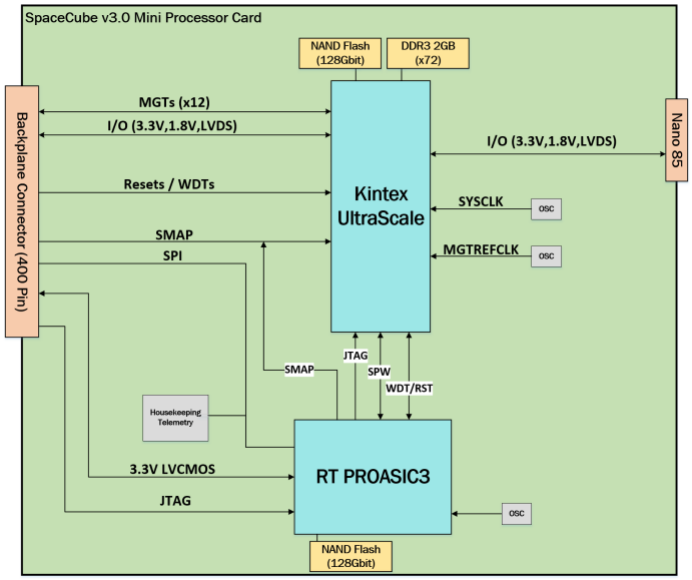

Figura 46: Schema a blocchi di SpaceCube v3.0 Mini. [27]

Le ragioni che hanno condotto alla scelta di utilizzare un FPGA Kintex UltraScale risiedono nell'ottima resistenza agli effetti dell'ambiente spaziale e nelle performance di elaborazione che tale dispositivo è in grado di offrire. Alcune delle tecniche di mitigazione prima presentate trovano diretta applicazione in questo progetto, infatti il MicroBlaze TMR e il SEM IP vengono utilizzati in quanto costituiscono degli strumenti fondamentali per raggiungere gli obiettivi di affidabilità richiesti. A tal proposito, gli sviluppatori del GSFC hanno deciso di utilizzare anche un supervisore esterno, l' RT PROASIC3, un FPGA realizzato con tecnologia FLASH a cui vengono affidate una serie di funzioni estremamente importanti. Esso svolge tali operazioni in modo indipendente dal resto della scheda e può rispondere ai comandi inviati da terra anche quando l'FPGA Kintex non è operativo. Il supervisore gestisce la configurazione dell'FPGA primario attraverso l'interfaccia SelectMap ed esegue lo scrubbing della memoria di configurazione. La scheda rappresentata nello schema a blocchi è stata progettata per poter essere facilmente inserita in schede madri più grandi per realizzare cluster contenenti molteplici elementi base connessi tra loro, come ad esempio altri SpaceCube v3.0 Mini. Tra le possibili applicazioni per cui questi moduli verranno impiegati vi sono algoritmi di machine learning, computer vision, osservazioni terrestri, algoritmi di compressione dati ed infine applicazioni di remote sensing.

5.2 Considerazioni Finali

In questo elaborato è stata proposta una panoramica delle principali caratteristiche che rendono gli FPGA Kintex UltraScale di Xilinx dei dispositivi adatti ad essere utilizzati in applicazioni spaziali. Questi dispositivi offrono un alto grado di efficienza sotto molteplici punti di vista. Le performance di calcolo sono migliori grazie ai moduli dedicati e pensati per applicazioni specifiche, come ad esempio i blocchi DSP. Il design sviluppato dal programmatore viene ottimizzato da appositi strumenti software per ridurre al minimo gli sprechi di superficie utile e per sfruttare al meglio le risorse disponibili. Tali accorgimenti sono necessari per poter soddisfare le richieste di un mercato in forte espansione quale il settore spaziale. Le caratteristiche che rendono gli FPGA Kintex UltraScale ideali alle applicazioni spaziali non si limitano alla capacità di calcolo, ma comprendono anche la resistenza alle difficili condizioni in cui sono chiamati ad operare. La presenza di radiazioni ionizzanti rappresenta una minaccia al funzionamento di un qualsiasi dispositivo elettronico. Uno studio attento dell'ambiente spaziale risulta di fondamentale importanza. Caratterizzare gli effetti negativi che le radiazioni possono avere sui dispositivi di interesse consente lo sviluppo di contromisure adatte per ridurre il verificarsi di situazioni indesiderate e di potenziale rischio per l'intera missione. Triplicazione, codici EDAC, scrubbing e riconfigurazione in orbita sono tra gli strumenti più utili per raggiungere i livelli di affidabilità richiesti in questo ambito.

Bibliografia

[1] Xilinx. (2022) Company overview. [Online]. Available: https://www.xilinx.com/about/company-overview.html

[2] ——. (2020) Rt kintex ultrascale fpgas for ultra high throughput and high bandwidth applications. [Online]. Available: https://docs.xilinx.com/v/u/en-US/wp523-xqrku060

[3] ——. (2014) Xilinx ultrascale: The next-generation architecture for your next-generation architecture. [Online]. Available: https://docs.xilinx.com/v/u/en-US/wp435-Xilinx-UltraScale

[4] ——. (2018) Understanding fpga architecture. [Online]. Available: https://www.xilinx.com/htmldocs/xilinx2017_4/sdaccel_doc/odz1504034293215.html

[5] ——. (2022) Getting started with vivado ip integrator. [Online]. Available: https://docs.xilinx.com/r/en-US/ug994-vivado-ip-subsystems/Getting-Started-with-Vivado-IP-Integrator

[6] ——. (2015) Xilinx ultrascale architecture for high-performance, smarter systems. [Online]. Available: https://docs.xilinx.com/v/u/en-US/wp434-ultrascale-smarter-systems

[7] ——. (2017) Ultrascale architecture clb. [Online]. Available: https://docs.xilinx.com/v/u/en-US/ug574-ultrascale-clb

[8] ——. (2021) Ultrascale architecture memory resources. [Online]. Available: https://docs.xilinx.com/v/u/en-US/ug573-ultrascale-memory-resources

[9] ——. (2021) Ultrascale architecture dsp slice. [Online]. Available: https://docs.xilinx.com/v/u/en-US/ug579-ultrascale-dsp

[10] ——. (2021) Ultrascale architecture clocking resources. [Online]. Available: https://docs.xilinx.com/v/u/en-US/ug572-ultrascale-clocking

[11] J. Heiner, B. Sellers, M. Wirthlin, and J. Kalb, "Fpga partial reconfiguration via configuration scrubbing," in *2009 International Conference on Field Programmable Logic and Applications*, 2009, pp. 99–104.

[12] Xilinx. (2022) Ultrascale architecture configuration. [Online]. Available: https://docs.xilinx.com/v/u/en-US/ug570-ultrascale-configuration

[13] E. E. S. A. Christian Poivey. (2019) Radiation effects in space electronics. [Online]. Available: https://indico.cern.ch/event/777129/contributions/3249529/attachments/1844695/3026130/6th_EIROforum_school_on_instrumentation_cpoivey.pdf

[14] H. J. Barnaby, "Total-ionizing-dose effects in modern cmos technologies," *IEEE Transactions on Nuclear Science*, vol. 53, no. 6, pp. 3103–3121, 2006.

[15] P. E. Dodd, M. R. Shaneyfelt, J. R. Schwank, and J. A. Felix, "Current and future challenges in radiation effects on cmos electronics," *IEEE Transactions on Nuclear Science*, vol. 57, no. 4, pp. 1747–1763, 2010.

[16] D. S. Lee, G. R. Allen, G. Swift, M. Cannon, M. Wirthlin, J. S. George, R. Koga, and K. Huey, "Single-event characterization of the 20 nm xilinx kintex ultrascale field-programmable gate array under heavy ion irradiation," in *2015 IEEE Radiation Effects Data Workshop (REDW)*, 2015, pp. 1–6.

[17] R. Ladbury, "Radiation hardening at the system level," in *IEEE NSREC Short Course*, 2007, pp. 1–94.

[18] G. Bruguier and J.-M. Palau, "Single particle-induced latchup," *IEEE Transactions on Nuclear Science*, vol. 43, no. 2, pp. 522–532, 1996.

[19] Xilinx. (2022) Radiation tolerant kintex ultrascale fpga data sheet. [Online]. Available: https://docs.xilinx.com/v/u/en-US/ds882-xqr-kintex-ultrascale

[20] B. Pratt, M. Caffrey, P. Graham, K. Morgan, and M. Wirthlin, "Improving fpga design robustness with partial tmr," in *2006 IEEE International Reliability Physics Symposium Proceedings*, 2006, pp. 226–232.

[21] C. Bolchini, A. Miele, and M. D. Santambrogio, "Tmr and partial dynamic reconfiguration to mitigate seu faults in fpgas," in *22nd IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems (DFT 2007)*, 2007, pp. 87–95.

[22] H. Quinn, K. Morgan, P. Graham, J. Krone, and M. Caffrey, "A review of xilinx fpga architectural reliability concerns from virtex to virtex-5," in *2007 9th European Conference on Radiation and Its Effects on Components and Systems*, 2007, pp. 1–8.

[23] Xilinx. (2016) Using the microblaze processor to accelerate cost-sensitive embedded system development. [Online]. Available: https://www.xilinx.com/content/dam/xilinx/support/documents/white_papers/wp469-microblaze-for-cost-sensitive-apps.pdf

[24] ——. (2022) Microblaze triple modular redundancy (tmr) subsystem. [Online]. Available: https://docs.xilinx.com/r/en-US/pg268-tmr/Triple-Modular-Redundancy-TMR-v1.0-LogiCORE-IP-Product-Guide-PG268

[25] A. Stoddard, A. Gruwell, P. Zabriskie, and M. J. Wirthlin, "A hybrid approach to fpga configuration scrubbing," *IEEE Transactions on Nuclear Science*, vol. 64, no. 1, pp. 497–503, 2017.

[26] Xilinx. (2021) Ultrascale architecture soft error mitigation controller v3.1 logicore ip product guide. [Online]. Available: https://docs.xilinx.com/r/en-US/pg187-ultrascale-sem/UltraScale-Architecture-Soft-Error-Mitigation-Controller-v3.1-LogiCORE-IP-Product-Guide

[27] C. Wilson, "Spacecube v3.0 mini nasa next-generation data-processing system for advanced cubesat applications," 2019. [Online]. Available: https://ntrs.nasa.gov/api/citations/20190027308/downloads/20190027308.pdf