MINISTERO DELLO SVILUPPO ECONOMICO
DIREZIONE GENERALE PER LA LOTTA ALLA CONTRAFFAZIONE
UFFICIO ITALIANO BREVETTI E MARCHI

DOMANDA DI INVENZIONE NUMERO	**102019000007371**
Data Deposito	**27/05/2019**
Data Pubblicazione	**27/11/2020**

Classifiche IPC

Sezione	Classe	Sottoclasse	Gruppo	Sottogruppo
G	06	F	11	10

Sezione	Classe	Sottoclasse	Gruppo	Sottogruppo
H	03	M	13	09

Titolo

Circuito di Cyclic Redundancy Check, dispositivo e procedimento corrispondenti

DESCRIZIONE dell'invenzione industriale dal titolo:
"Circuito di Cyclic Redundancy Check, dispositivo e procedimento corrispondenti"
di: STMicroelectronics S.r.l., nazionalità italiana, Via C. Olivetti, 2 - 20864 Agrate Brianza (MB)
Inventore designato: Luca SASSELLI
Depositata il: 27 maggio 2019

****

TESTO DELLA DESCRIZIONE

Campo tecnico

La descrizione si riferisce alle tecniche che prevedono il calcolo di Cyclic Redundancy Check, in breve CRC.

Una o più forme di attuazione possono essere applicate, per esempio, alla verifica dell'integrità dei dati, per esempio, nella trasmissione e memorizzazione di dati.

Sfondo tecnologico

Varie reti digitali e dispositivi di memorizzazione adottano un Cyclic Redundancy Check, in breve CRC, come codice per la rilevazione di errori allo scopo di rilevare cambiamenti accidentali nei dati.

Quando si utilizza il CRC, ai blocchi dei dati che entrano in questi sistemi viene associato un breve valore di controllo, in base al resto di una divisione polinomiale dei loro contenuti. Un approccio comune applicato nei circuiti per il calcolo del CRC prevede la produzione di CRC come il resto di una divisione polinomiale che prevede dati di ingresso e un divisore fisso, indicato come il

polinomio CRC. I circuiti che implementano il CRC sono utilizzati in modo esteso in un'ampia varietà di applicazioni e standard.

Scopo e sintesi

Nonostante l'intensa attività in quest'area, una aumentata flessibilità e riusabilità di un circuito CRC rappresenta caratteristiche desiderabili.

Uno scopo di una o più forme di attuazione è di contribuire a fornire in modo corrispondente soluzioni perfezionate.

Secondo una o più forme di attuazione, tale scopo può essere raggiunto per mezzo di un circuito che presenta le caratteristiche esposte nelle rivendicazioni che seguono.

Una o più forme di attuazione possono riferirsi a un dispositivo corrispondente.

Una o più forme di attuazione possono riferirsi a un procedimento corrispondente, per esempio per l'impiego nel protocollo SPI.

Le rivendicazioni sono una parte integrante della divulgazione dell'invenzione come qui fornita.

Una o più forme di attuazione facilitano il calcolo di un Cyclic Redundancy Check (CRC, verifica di ridondanza ciclica) che prevede il calcolo del checksum di un dataset a N bit in N cicli di clock, in assenza di limitazioni di seed.

Una o più forme di attuazione possono fornire vantaggi in termini di cicli di clock in confronto a un circuito CRC seriale (un registro a scorrimento a retroazione lineare o LFSR, "Linear-Feedback Shift Register", per esempio) e in termini di area di semiconduttore in confronto a un

circuito CRC combinatorio, dando luogo così ad una sintesi migliorativa.

Una o più forme di attuazione possono agevolare le applicazioni di protocolli di trasmissione seriale (protocolli SPI, per esempio).

Una o più forme di attuazione possono prevedere una specie di architettura ibrida seriale/combinatoria che prevede una elaborazione sia seriale che parallela.

Breve descrizione delle figure

Una o più forme di attuazione verranno adesso descritte, solo a titolo di esempio, con riferimento alle figure allegate, in cui:

la figura 1A è esemplificativa di un circuito che implementa una soluzione seriale per calcolare un codice CRC,

la figura 1B è esemplificativa di un circuito che implementa una soluzione combinatoria per calcolare un codice CRC,

la figura 2A è un diagramma circuitale esemplificativo di un circuito che implementa un calcolo seriale di CRC,

la figura 2B è un altro diagramma circuitale esemplificativo di un circuito che implementa un calcolo seriale di CRC, e

la figura 3A è un diagramma circuitale esemplificativo di un circuito che implementa un calcolo parallelo (combinatorio) di CRC,

la figura 3B è esemplificativa di un possibile blocco costruttivo in un circuito come esemplificato nella figura 3A,

la figura 4 è un diagramma circuitale esemplificativo

di un circuito secondo forme di attuazione, e

la figura 5 è un diagramma più dettagliato di una possibile implementazione di un diagramma circuitale come esemplificato nella figura 4.

Descrizione dettagliata di forme di attuazione esemplificative

Nella descrizione che segue sono illustrati uno o più dettagli specifici, rivolti a fornire una comprensione approfondita di esempi di forme di attuazione. Le forme di attuazione possono essere ottenute senza uno o più dei dettagli specifici, o con altri procedimenti, componenti, materiali, ecc. In altri casi, strutture, materiali, o operazioni noti non sono illustrati o descritti in dettaglio cosicché certi aspetti di forme di attuazione non verranno offuscati.

Il riferimento a "una forma di attuazione" o "una sola forma di attuazione" nel quadro della presente descrizione è inteso a indicare che una particolare configurazione, struttura, o caratteristica descritta in relazione alla forma di attuazione è compresa in almeno una forma di attuazione. Quindi, frasi come "in una forma di attuazione" o "in una sola forma di attuazione" che possono essere presenti in uno o più punti della presente descrizione non si riferiscono necessariamente a una specifica forma di attuazione. Inoltre, particolari conformazioni, strutture, o caratteristiche possono essere combinate in qualsiasi modo adeguato in una o più forme di attuazione.

I riferimenti qui utilizzati sono forniti solamente per comodità e quindi non definiscono l'estensione di protezione o la portata delle forme di attuazione.

Come ben noto per gli esperti del ramo, l'implementazione di CRC avente una certa "larghezza" (K bit, per esempio) può prevedere l'impiego di un polinomio che presenta un valore di "seed" (seme) per inizializzare il circuito.

Una discussione generale dei principi di base e dei criteri sottostanti all'implementazione di CRC può essere trovata in Peterson, W. W. e Brown, D. T.: "Cyclic Codes for Error Detection", Proceedings of the IRE, gennaio 1961, pp. 228 - 235.

Le figure 1A e 1B sono esemplificative di circuiti (convenzionali) configurati per calcolare un CRC a K bit (cioè comprendente K bit) utilizzando un seed di K bit a partire da una stringa di N bit implementando una soluzione seriale S (figura 1A, con bit di ingresso D_0, …, D_{N-1} forniti in modo seriale a un ingresso di circuito In) o una soluzione parallela P (figura 1B, con un insieme di bit di ingresso collettivamente denominati D forniti in parallelo all'ingresso di circuito In).

I circuiti seriali (figura 1A, per esempio) presentano un'impronta di area più piccola con lo svantaggio di più cicli di clock coinvolti nel calcolo del risultato.

I circuiti paralleli o combinatori (figura 1B, per esempio) presentano il vantaggio di calcolare il risultato direttamente da un intero dataset di ingresso D con svantaggi correlati a consumo di risorse, velocità inferiore e riusabilità limitata per dimensioni variabili dei dati (dei dati di ingresso D).

La figura 2A è esemplificativa di una struttura di circuito seriale che può essere utilizzato per calcolare un CRC da una stringa di bit di ingresso D_0, …, D_{N-1} forniti

modo seriale a un ingresso di circuito utilizzando un registro a scorrimento con retroazione lineare (in breve LFSR). La soluzione (convenzionale) illustrata nella figura 2A può essere considerata come esemplificativa in qualche modo di una architettura "non-forward".

Il diagramma della figura 2A è esemplificativo di una struttura che implementa un calcolo di CRC sulla base di un polinomio che può essere espresso come:

$$1+x+x^2+x^3+x^5+x^8$$

La struttura come esemplificata nella figura 2A comprende 8 registri (flip-flop, FF0 a FF7) in una struttura in cascata con i registri FF0, FF1, FF2, FF3 e FF5 preceduti da rispettive porte OR esclusivo (EX-OR o XOR) XOR0, XOR1, XOR2, XOR3 e XOR5. I registri FF0, FF1, FF2, FF3 e FF5 nella struttura in cascata sono disposti in posizioni che corrispondono agli esponenti degli elementi del polinomio CRC con $x^0=1$ con l'eccezione del termine di ordine più elevato nel polinomio, cioè x^8.

Il gate XOR0 riceve i dati di ingresso (seriali) in uno dei suoi ingressi e ciascuna delle altre porte XOR, cioè XOR1, XOR2, XOR3 e XOR5 riceve in uno dei suoi ingressi l'uscita dal registro precedente nella struttura in cascata (cioè XOR1 da FF0, XOR2 da FF1, XOR3 da FF2 e XOR5 da FF4).

Tutte le porte XOR ricevono nei loro altri ingressi un segnale di retroazione dall'uscita dell'ultimo registro nella struttura in cascata (cioè FF7).

In una struttura come esemplificata nella figura 2A, configurata per implementare il polinomio CRC $1+x+x^2+x^3+x^5+x^8$ come discusso in precedenza, i registri FF4, FF6 e FF7 (cioè quei registri che non sono preceduti da una

porta XOR) ricevono il loro ingresso dall'uscita del registro precedente (cioè FF4 da FF3, FF6 da FF5 e FF7 da FF6).

Una struttura come esemplificata nella figura 2A (per esempio in relazione alle possibili posizioni delle porte XOR nelle posizioni dettate dal polinomio CRC come discusso in precedenza) è d'altra parte convenzionale nella tecnica, il che rende superfluo fornire in questa sede una descrizione di maggior dettaglio.

Il funzionamento di un circuito seriale come esemplificato nella figura 2A per calcolare un CRC prevede l'inizializzazione dei registri con i valori di "seed", facendo scorrere i dati nello LFSR seguiti da una quantità di "zeri" uguale alla larghezza del CRC.

Il funzionamento di un circuito seriale come esemplificato nella figura 2A per calcolare un CRC prevede l'inizializzazione dei registri con i valori di "seed" con i dati sottoposti a scorrimento (in un modo ciclico nel registro a scorrimento a retroazione lineare) con una quantità di "zeri" uguale alla larghezza del CRC sottoposto a scorrimento nello LFSR.

Il funzionamento di una struttura come esemplificata nella figura 2A prevede lo scorrimento di K zeri dopo gli N bit di dati. Calcolare un CRC largo K bit da un dataset a N bit prevede quindi K + N cicli di clock prodotti da un generatore di clock (non visibile nella figura).

Tale approccio può essere svantaggioso per vari motivi.

In primo luogo, può essere richiesta una circuiteria logica aggiuntiva nella gestione di tale compito nella misura in cui un circuito come esemplificato nella figura

2A non sia in grado di fornire da solo una prestazione di calcolo desiderata.

Inoltre, la struttura come esemplificata nella figura 2A rende il processo di calcolo del CRC dipendente dalla dimensione del dataset. Di conseguenza, devono essere fornite informazioni precise alla circuiteria logica responsabile per lo scorrimento degli zeri sulla quantità di dati per i quali il checksum è calcolato.

Ancora, il funzionamento di una struttura come esemplificata nella figura 2A dà luogo a una separazione tra l'istante in cui l'ultimo bit di dati viene fatto scorrere e l'istante in cui il CRC viene effettivamente calcolato.

In applicazioni come i protocolli di trasmissione seriale (interfaccia periferica seriale o SPI, "Serial Peripheral Interface", per esempio) dove il checksum viene inviato dopo i dati, una struttura come esemplificata nella figura 2A depone contro il calcolo del CRC quando il carico utile viene trasmesso.

La capacità di rimuovere tale vincolo può favorire la flessibilità e la riusabilità del circuito CRC.

La figura 2B è esemplificativa di una struttura derivata dalla struttura della figura 2A in cui, sotto l'ipotesi di un seed di CRC uguale a zero, viene fornita la possibilità di calcolare il CRC in N cicli di clock.

Nella figura 2B, parti o elementi simili a parti o elementi già discussi in relazione alla figura 2A sono indicati con simboli di riferimento simili: una descrizione dettagliata corrispondente della struttura della figura 2B non verrà ripetuta a scopo di brevità.

Essenzialmente (nell'ipotesi che la struttura della

figura 2B condivida con la struttura della figura 2A lo stesso polinomio CRC, cioè $1+x+x^2+x^3+x^5+x^8$), in una struttura come esemplificata nella figura 2B i dati di ingresso D_0, …, D_{N-1} sono (anche) forniti a un (terzo) ingresso per le porte XOR XOR1, XOR2, XOR3 e XOR5.

Va notato che l'utilizzo di un seed "nullo" (con un seed di CRC uguale a zero) può rappresentare una limitazione nella misura in cui i) solo certi codici CRC possono essere compatibili con tale specifica, ii) certi standard specificano valori di seed multipli: si veda, per esempio,

http://reveng.sourceforge.net/crc-catalogue/all.htm.

In aggiunta, si osserva che, anche se di per sé ammissibile, un seed nullo può essere meno vantaggioso di un seed non nullo. Di fatto, se il circuito viene inizializzato con un valore di seed pari a zero, il CRC di un dataset di ingresso nullo sarà in modo simile nullo, il che depone contro la rilevazione di guasti "bloccato a zero".

Il diagramma della figura 3A è esemplificativo di una struttura che implementa un calcolo di CRC (nuovamente) facendo ricorso a un approccio parallelo (combinatorio) sulla base di un polinomio che può essere espresso come:

$$1+x+x^2+x^3+x^5+x^8$$

facendo ricorso a un approccio parallelo (combinatorio).

La struttura come esemplificata nella figura 3A comprende due insiemi di otto "blocchi costruttivi" BB0 a BB7 e BB0’ a BB7’ in due disposizioni in cascata, in cui:

- nella prima struttura (BB0 a BB7), BB0 riceve un seed di K bit come ingresso e ciascuno di BB1 a BB7 riceve

come ingresso IN l'uscita OUT dal precedente blocco costruttivo nella struttura in cascata con i bit di dati D0 a D7 (dati D in formato parallelo) applicati come rispettivi ingressi a BB0 a BB7; e

- nella seconda struttura (BB0' a BB7'), BB0' riceve come ingresso l'uscita di K bit dall'ultimo blocco costruttivo della prima struttura (BB0 a BB7) e ciascuno di BB1' a BB7' riceve come ingresso IN l'uscita OUT dal blocco costruttivo precedente con i rispettivi ingressi (di dati) tutti impostati a zero ("0").

Come risultato, il numero di blocchi costruttivi aumenta con la dimensione dei dati.

Ciascuno dei blocchi costruttivi BBj e BBj' con j = 0, …, 7 esemplificati nella figura 3A può essere configurato come esemplificato nella figura 3B, cioè come comprendente un circuito multiplexer 100 che presenta due ingressi (contrassegnati con "0" e "1" nella figura 3B) configurati per ricevere:

- una replica con scorrimento di 1 bit dell'ingresso IN (a k bit) da un circuito a scorrimento 102, e

- l'uscita da una porta OR esclusivo XORj, rispettivamente.

La porta OR esclusivo XORj a sua volta riceve:

- come primo ingresso, la replica con scorrimento di 1 bit dell'ingresso IN (a k bit) dal circuito a scorrimento 102, e

- come secondo ingresso, una versione codificata del polinomio CRC che è una stringa di bit dove gli "uni" nella stringa corrispondono agli esponenti degli elementi del polinomio CRC, con $x^0=1$ (cioè, nella struttura in cascata sono disposti in posizioni che corrispondono agli esponenti

degli elementi del polinomio CRC con $x^0=1$ con l'eccezione del termine di ordine più elevato nel polinomio, cioè x^8): per esempio 11110100 per $1+x+x^2+x^3+x^5+x^8$, dove gli "zeri" nella stringa di bit della versione codificata del polinomio CRC corrispondono a x^4, x^6, x^7, cosicché il secondo ingresso della porta OR esclusivo XORj sarà "0" per XORj con j = 4, 6 and 7.

Come esemplificato nella figura 3A, il multiplexer 100 può essere controllato (allo scopo di trasferire all'uscita OUT uno degli ingressi contrassegnati con "0" and "1") in funzione di un segnale a 1 bit SELj corrispondente ai dati D, cioè Selj = Dj con j = 0, …, 7.

Nuovamente, la struttura come esemplificata nelle figure 3A e 3B è convenzionale nella tecnica, il che rende superfluo fornire in questa sede una descrizione di maggior dettaglio.

Si osserva d'altra parte che impostare a "0" il segnale a 1 bit SELj corrispondente ai dati D nella parte inferiore della figura 3A (cioè Selj = 0 con j = 0, …, 7 per tutti i blocchi BB0' a BB7') corrisponde a impostare il multiplexer 100 in esso a una posizione fissa cosicché tale multiplexer, pur essendo illustrato per facilità di spiegazione, può essere di fatto eliminato.

Come discusso in precedenza, i circuiti paralleli o combinatori (figure 3A e 3B, per esempio) possono presentare svantaggi correlati a consumo di risorse, velocità inferiore e riusabilità limitata per dimensioni variabili dei dati (dei dati di ingresso D).

Una o più forme di attuazione possono basarsi su un'architettura di circuito ibrido come esemplificata nella figura 4.

Tale circuito ibrido (indicato complessivamente con 10) combina:

- un circuito CRC seriale 12 (essenzialmente come esemplificato nella figura 1A e discusso più in dettaglio in relazione alla figura 2A), e
- un circuito CRC parallelo/combinatorio 14 (essenzialmente come esemplificato nella figura 1B e discusso più in dettaglio in relazione alle figure 3A e 3B).

In una struttura ibrida come esemplificata nella figura 4 i dati di ingresso D_0, …, D_{N-1} (come forniti da un circuito sorgente o di ingresso U) possono essere forniti in modo seriale a un nodo di ingresso 120 del circuito seriale 12 che riceve inoltre un seed di K bit fornito (in qualsiasi modo noto agli esperti nel ramo) da un ingresso di seed 122.

In una struttura come esemplificata nella figura 4, l'elaborazione CRC seriale nel circuito 12 darà luogo a un segnale "intermedio" (a K bit) fornito in un nodo di uscita 124 che può essere fornito come seed a un nodo di ingresso di seed 142 del circuito CRC parallelo 14 che è configurato per ricevere dati nulli ("zeri") nel suo nodo di ingresso di K bit 140 per produrre in un nodo di uscita 144 un CRC finale desiderato (a K bit) da fornire a un circuito utilizzatore o di uscita OU.

In una o più forme di attuazione un insieme del circuito 10 più il circuito sorgente o di ingresso IU e del circuito utilizzatore o di uscita OU può essere incluso in qualsiasi dispositivo configurato per implementare una procedura di checksum, per verificare l'integrità dei dati, per esempio, nella trasmissione e memorizzazione di dati,

per esempio.

Per esempio, l'unità sorgente o di ingresso IU può trasmettere su un "canale" C dati la cui integrità viene verificata utilizzando il CRC reso disponibile nel circuito utilizzatore o di uscita OU.

Per esempio, una tale struttura può essere utilizzata per verificare l'integrità di dati come memorizzati in una memoria alla lettura di questi dati dalla memoria, per esempio.

In una o più forme di attuazione come esemplificate nella figura 4, lo stadio seriale 12 può comprendere un circuito CRC seriale (non forward) come esemplificato nella figura 2A e lo stadio parallelo 14 può comprendere un circuito CRC combinatorio come esemplificato nelle figure 3A e 3B che funziona sulla base dello stesso polinomio con:

- l'uscita dallo stadio seriale 12 nel nodo 124 fornita come seed allo stadio parallelo/combinatorio 14 nel nodo di ingresso di seed 142, e
- l'ingresso per lo stadio combinatorio 14 nell'ingresso 140 impostato a zero (K zeri).

Utilizzare l'uscita (nodo 124) dallo stadio seriale 12 come seed per lo stadio combinatorio 14 (nodo 142) fa sì che lo stadio combinatorio 14 calcoli il risultato dello scorrimento di K zeri a partire dal risultato dello stadio seriale.

La figura 5 è una rappresentazione più dettagliata di una possibile implementazione di un circuito corrispondente alla struttura esemplificata in generale nella figura 4.

La rappresentazione della figura 5 è esemplificativa della possibilità di ottimizzare tale implementazione oltre il semplice assemblaggio di una implementazione come

esemplificata nella figura 3A (elaborazione CRC "parallela" o "combinatoria") con una implementazione come esemplificata nella figura 2A (elaborazione CRC "seriale").

Nella figura 5, parti o elementi simili a parti o elementi già discussi in relazione alla figura 4 (e anche alle altre figure) sono indicati con simboli di riferimento simili: una descrizione dettagliata corrispondente della struttura della figura 5 non verrà ripetuta a scopo di brevità.

Brevemente, la figura 5 è una rappresentazione più dettagliata di un circuito esemplificato nella figura 4 implementato utilizzando, per il primo stadio di circuito 12 (configurato per implementare una elaborazione CRC seriale) una struttura come esemplificata nella figura 2A che riceve i bit di dati di ingresso D_0, …, D_{N-1} (ingresso 120) e il seed di CRC (ingresso 122) e produce nella porta di uscita 124 dati di elaborazione CRC "intermedi" in funzione dei bit di dati di ingresso e del seed di CRC.

Come esemplificato nella figura 5, questi dati di CRC elaborati "intermedi" possono essere forniti come un seed di CRC all'ingresso di seed 142 del secondo stadio di circuito 14 (configurato per implementare una elaborazione CRC parallela o combinatoria) proprio come l'uscita dalla struttura "superiore" nella figura 3A (BB0 a BB7) è applicata alla seconda struttura (BB0' a BB7') con i rispettivi ingressi (di dati) tutti impostati a zero ("0", bit di dati di ingresso nulli) nell'ingresso 140.

Come qui esemplificato, il secondo stadio di circuito 14 è configurato per implementare una elaborazione CRC parallela (combinatoria) dei bit di dati di ingresso nulli D = "0" e dei dati intermedi ricevuti (dalla linea 124 nel

primo stadio di circuito 12) nella rispettiva porta di ingresso di seed 142. I dati elaborati risultanti di CRC sono prodotti nella rispettiva porta di uscita 144 in funzione dei bit di dati di ingresso nulli e dei dati di CRC elaborati intermedi.

Qui nuovamente, si osserva che impostare a "0" il segnale a 1 bit SELj corrispondente ai dati D nella parte inferiore della figura 5 (cioè Selj = 0 con j = 0, …, 7 per tutti i blocchi BB0′ a BB7′) corrisponde a impostare il multiplexer 100 (si veda la rappresentazione nella figura 3B) a una posizione fissa cosicché tale multiplexer può di fatto essere eliminato in tutti i blocchi BB0′ to BB7′.

In tale modo, il nodo di uscita OUT in ciascun blocco BB0′ to BB7′ riceverà direttamente l'uscita dalla porta X-OR XORj che presenta:

- un ingresso accoppiato al nodo di ingresso IN attraverso lo scorrimento di 1 bit 102, e
- l'altro ingresso configurato per ricevere il segnale ENC POLY, cioè uno dei bit in una stringa che fornisce una versione codificata del polinomio CRC, cioè una stringa di bit in cui gli "uni" nella stringa corrispondono agli esponenti degli elementi del polinomio CRC, $x^0=1$ essendo disposti in posizioni nella stringa che corrispondono agli esponenti degli elementi del polinomio CRC con $x^0=1$ (con l'eccezione del termine di ordine più elevato nel polinomio, cioè x^8): per esempio 11110100 per $1+x+x^2+x^3+x^5+x^8$.

In una o più forme di attuazione come esemplificate nelle figure 4 e 5, il CRC di un dataset di N bit (D_0, …, D_{N-1}) può essere calcolato in N cicli di clock (da un generatore di clock CLK) senza limitazioni di rilievo sulla

selezione del seed applicato all'ingresso di seed 122 dello stadio seriale 12. Una o più forme di attuazione come esemplificate nella figura 4 possono così combinare i vantaggi delle architetture forward e non forward, come esemplificate nelle figure 2A e 2B.

Una o più forme di attuazione come esemplificate nelle figure 4 e 5 facilitano il calcolo del CRC di un dataset di N bit in N cicli di clock. Quando l'ultimo bit viene fatto scorrere nell'ingresso 120 il CRC è pronto con l'uscita (in 144 nella figura 4) contenente una CRC valido per i dati sottoposti a scorrimento fino a quel punto.

Di conseguenza, in quelle applicazioni come i protocolli di trasmissione seriale in cui il checksum viene inviato (su un canale come ad esempio C nella figura 4, per esempio) dopo i dati, una struttura come esemplificata nelle figure 4 e 5 facilita il calcolo del CRC quando il carico utile viene trasmesso.

Una o più forme di attuazione come esemplificate nelle figure 4 e 5 prevedono una quantità ridotta di logica esterna per l'integrazione.

Inoltre una o più forme di attuazione come esemplificate nelle figure 4 e 5 facilitano l'impiego virtualmente di qualsiasi seed virtualmente con qualsiasi tipo di CRC atto a essere implementato.

Una o più forme di attuazione possono ottenere questi vantaggi con un aumento tollerabile di sovraccarico di area/ritardo principalmente correlato allo stadio combinatorio 14. D'altra parte si osserva che tale sovraccarico è proporzionale alla larghezza del CRC (K, per esempio) e non rispetto alla dimensione del dataset (N, per esempio) come nel caso di architetture completamente

combinatorie, essendo quindi trascurabile per piccole larghezze di CRC.

Un circuito (per esempio, 10) per calcolare un (codice di) cyclic redundancy check, correntemente indicato come CRC, come qui esemplificato può comprendere:

- un primo stadio di circuito (per esempio, 12, S) che presenta un nodo di ingresso di dati seriali (per esempio, 120) configurato per ricevere bit di dati di ingresso (per esempio, D_0, …, D_{N-1}), una porta di ingresso di seed (per esempio, 122) configurata per ricevere un seed di CRC e una porta di uscita (per esempio, 124), il primo stadio di circuito configurato per implementare una elaborazione di CRC seriale di detti bit di dati di ingresso e detto seed di CRC e produrre dati elaborati intermedi di CRC in detta porta di uscita in funzione di detti bit di dati di ingresso e detto seed di CRC, e

- un secondo stadio di circuito (per esempio, 14, P) che presenta una rispettiva porta di ingresso di seed (per esempio, 142) accoppiata alla porta di uscita del primo stadio di circuito per ricevere da essa detti dati elaborati intermedi di CRC e una rispettiva porta di uscita (per esempio, 144), il secondo stadio di circuito configurato per implementare una elaborazione CRC parallela di bit di dati di ingresso nulli e detti dati elaborati intermedi di CRC ricevuti in detta rispettiva porta di ingresso di seed e produrre dati elaborati risultanti di CRC in detta rispettiva porta di uscita in funzione di detti bit di dati di ingresso nulli e detti dati di CRC elaborati intermedi.

In un circuito come qui esemplificato, il primo stadio di circuito e il secondo stadio di circuito possono essere

configurati per implementare una elaborazione CRC seriale e una elaborazione CRC parallela, rispettivamente, in funzione dello stesso polinomio CRC.

In un circuito come qui esemplificato, il primo stadio di circuito può comprendere un registro a scorrimento a retroazione lineare, LFSR.

In un circuito come qui esemplificato (si vedano per esempio la figura 5 e la figura 3B, in cui il multiplexer 100 può essere eliminato se SELj = 0, cioè con dati di ingresso nulli):

- il secondo stadio di circuito (14, P) può comprendere una struttura in cascata di blocchi combinatori (per esempio BB0', …, BB7'), i blocchi combinatori presentando un nodo di ingresso (per esempio, IN) e un nodo di uscita (per esempio, OUT) con una porta OR esclusivo (per esempio, XORj) che presenta un rispettivo nodo di uscita accoppiato al nodo di uscita, un primo rispettivo ingresso accoppiato al nodo di ingresso attraverso un blocco a scorrimento a 1 bit (per esempio, 102) e un secondo rispettivo ingresso (per esempio, ENC POLY) configurato per ricevere uno dei bit in una stringa che fornisce una versione codificata del polinomio CRC,

- il nodo di ingresso del primo blocco nella struttura in cascata dei blocchi combinatori può essere accoppiato alla porta di uscita del primo stadio di circuito per ricevere da esso detti dati elaborati intermedi di CRC,

- i blocchi nella struttura in cascata dei blocchi combinatori diversi da detto primo blocco possono presentare il loro nodo di ingresso accoppiato al nodo di uscita del blocco precedente nella struttura in cascata di blocchi combinatori,

- il nodo di uscita dell'ultimo blocco nella struttura in cascata di blocchi combinatori può far sì che detta rispettiva porta di uscita produca i dati elaborati risultanti di CRC in funzione di detti bit di dati di ingresso nulli e detti dati elaborati intermedi di CRC.

Un dispositivo come qui esemplificato (per esempio, IU, 10, OU) può comprendere:

- un circuito sorgente (per esempio IU) di detti bit di dati di ingresso,

- un circuito come qui esemplificato che presenta detto nodo di ingresso di dati seriali del primo stadio di circuito accoppiato a detto circuito sorgente per ricevere da esso detti bit di dati di ingresso, e

- un circuito utilizzatore (per esempio, OU) accoppiato a detta rispettiva porta di uscita del secondo stadio di circuito per ricevere da esso detti dati elaborati risultanti di CRC.

Un procedimento per trasferire bit di dati su un canale di trasferimento (per esempio, C) può comprendere fornire detti bit di dati al nodo di ingresso di dati seriali del primo stadio di circuito in un circuito (per esempio, 10) come qui esemplificato e produrre i dati elaborati risultanti di CRC in detta rispettiva porta di uscita durante il trasferimento di detti bit di dati di ingresso su detto canale di trasferimento.

Un procedimento come qui esemplificato può comprendere trasferire detti dati elaborati risultanti di CRC su detto canale di trasferimento successivamente al trasferimento di detti bit di dati di ingresso (come carico utile, per esempio) su detto canale di trasferimento.

Senza pregiudizio per i principi sottostanti, i

dettagli e le forme di attuazione possono variare, anche significativamente, rispetto a ciò che è stato descritto solo a titolo di esempio, senza allontanarsi dalla portata di protezione.

L'estensione di protezione è definita dalle rivendicazioni allegate.

RIVENDICAZIONI

1. Circuito (10) per calcolare un cyclic redundancy check, CRC, il circuito comprendendo:

- un primo stadio di circuito (12, S) che presenta un nodo di ingresso di dati seriali (120) configurato per ricevere bit di dati di ingresso (D_0, …, D_{N-1}), una porta di ingresso di seed (122) configurata per ricevere un seed di CRC e una porta di uscita (124), il primo stadio di circuito (12, S) configurato per implementare una elaborazione CRC seriale di detti bit di dati di ingresso (D_0, …, D_{N-1}) e detto seed di CRC e produrre dati intermedi elaborati di CRC in detta porta di uscita (124) in funzione di detti bit di dati di ingresso (D_0, …, D_{N-1}) e detto seed di CRC, e

- un secondo stadio di circuito (14, P) che presenta una rispettiva porta di ingresso (142) accoppiata alla porta di uscita (124) del primo stadio di circuito (12, S) per ricevere da esso detti dati intermedi elaborati di CRC e una rispettiva porta di uscita (144), il secondo stadio di circuito (14, P) configurato per implementare una elaborazione CRC parallela di bit di dati di ingresso nulli (0) e detti dati elaborati intermedi di CRC ricevuti in detta porta di ingresso di seed (142) e produrre dati elaborati risultanti di CRC in detta rispettiva porta di uscita (144) in funzione di detti bit di dati di ingresso nulli (0) e detti dati elaborati intermedi di CRC.

2. Circuito (10) secondo la rivendicazione 1 o la rivendicazione 2, in cui il primo stadio di circuito (12, S) e il secondo stadio di circuito (14, P) sono configurati per implementare una elaborazione CRC seriale e una

elaborazione CRC parallela, rispettivamente, in funzione di uno stesso polinomio CRC.

3. Circuito (10) secondo la rivendicazione 1 o la rivendicazione 2, in cui il primo stadio di circuito (12, S) comprende un registro a scorrimento a retroazione lineare, LFSR.

4. Circuito 10 secondo una qualsiasi delle precedenti rivendicazioni, in cui il secondo stadio di circuito (14, P) comprende un circuito CRC combinatorio.

5. Circuito (10) secondo la rivendicazione 4, in cui:

- il secondo stadio di circuito (14, P) comprende una struttura in cascata di blocchi combinatori (BB0', …, BB7'), i blocchi combinatori presentando un nodo di ingresso (IN) e un nodo di uscita (OUT) con una porta OR esclusivo (XORj) che presenta un rispettivo nodo di uscita accoppiato al nodo di uscita (OUT), un primo rispettivo nodo di ingresso accoppiato al nodo di ingresso (IN) attraverso un blocco a scorrimento a 1 bit (102), e un secondo rispettivo ingresso (ENC POLY) configurato per ricevere uno dei bit in una stringa che fornisce una versione codificata del polinomio CRC,

- il nodo di ingresso (IN) del primo blocco (BB0') nella struttura in cascata di blocchi combinatori (BB0', …, BB7') è accoppiato alla porta di uscita (124) del primo stadio di circuito (12, S) per ricevere da esso detti dati elaborati intermedi di CRC,

- i blocchi (BB1', …, BB7') nella struttura in cascata di blocchi combinatori diversi da detto primo blocco (BB0') presentano il loro nodo di ingresso (IN) accoppiato al nodo di uscita (OUT) nel blocco precedente (BB0', …, BB6') nella struttura in cascata di blocchi combinatori (BB0', …,

BB7'),

- il nodo di uscita (OUT) dell'ultimo blocco (BB7') nella struttura in cascata di blocchi combinatori (BB0', …, BB7') fornisce detta rispettiva porta di uscita (144) per produrre dati elaborati risultanti di CRC in funzione di detti bit di dati di ingresso nulli (0) e detti dati elaborati intermedi di CRC.

6. Dispositivo (IU, 10, OU) comprendente:

- un circuito sorgente (IU) di detti bit di dati di ingresso (D_0, …, D_{N-1}),

- un circuito secondo una qualsiasi delle rivendicazioni 1 a 5 che presenta detto nodo di ingresso di dati seriali (120) del primo stadio di circuito (12, S) accoppiato a detto circuito sorgente per ricevere da esso detti bit di dati di ingresso (D_0, …, D_{N-1}), e

- un circuito utilizzatore (OU) accoppiato a detta rispettiva porta di uscita (144) del secondo stadio di circuito (14, P) per ricevere da esso detti dati elaborati risultanti di CRC.

7. Procedimento per il trasferimento di bit di dati (D_0, …, D_{N-1}) su un canale di trasferimento (C) il procedimento comprendendo fornire detti bit di dati (D_0, …, D_{N-1}) al nodo di ingresso di dati seriali (120) del primo stadio di circuito (12, S) in un circuito (10) secondo una qualsiasi delle rivendicazioni 1 a 5 e produrre dati elaborati risultanti di CRC in detta rispettiva porta di uscita (144) durante il trasferimento di detti bit di dati di ingresso (D_0, …, D_{N-1}) su detto canale di trasferimento (C).

8. Procedimento secondo la rivendicazione 7, comprendente trasferire detti dati elaborati risultanti di

CRC su detto canale di trasferimento (C) successivamente al trasferimento di detti bit di dati di ingresso (D_0, …, D_{N-1}) su detto canale di trasferimento (C).

FIG. 1A

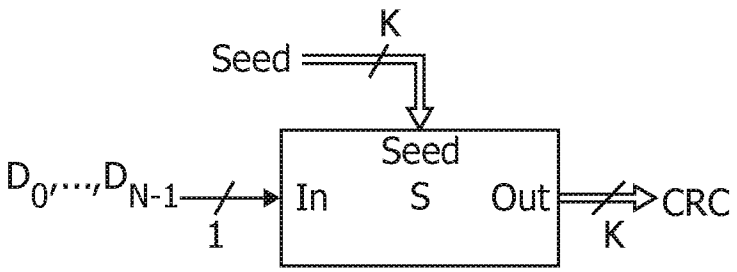


FIG. 1B

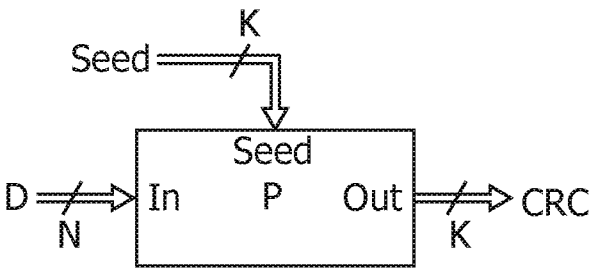


FIG. 2A

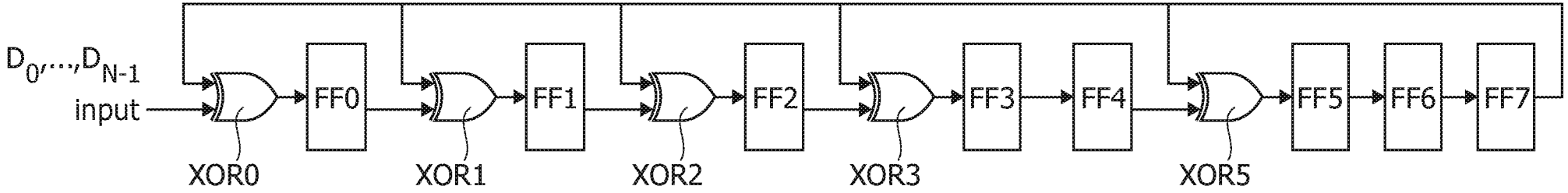


FIG. 2B

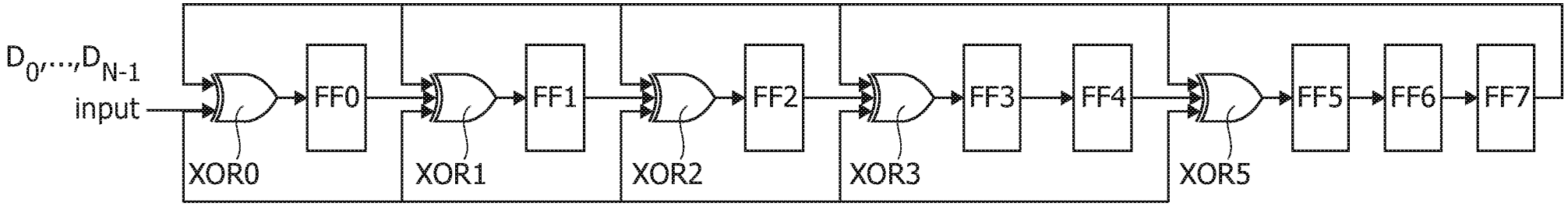


FIG. 3A

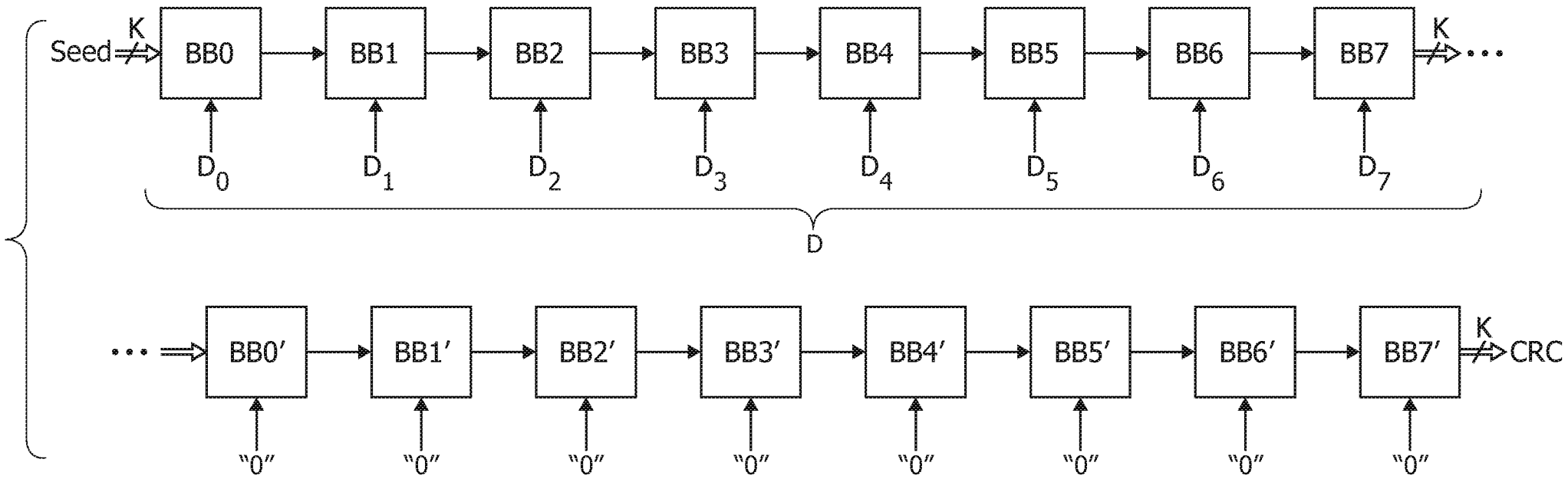

Seed
K
BB0
BB1
BB2
BB3
BB4
BB5
BB6
BB7
K
D0
D1
D2
D3
D4
D5
D6
D7
D
BB0′
BB1′
BB2′
BB3′
BB4′
BB5′
BB6′
BB7′
K
CRC
"0"

FIG. 3B

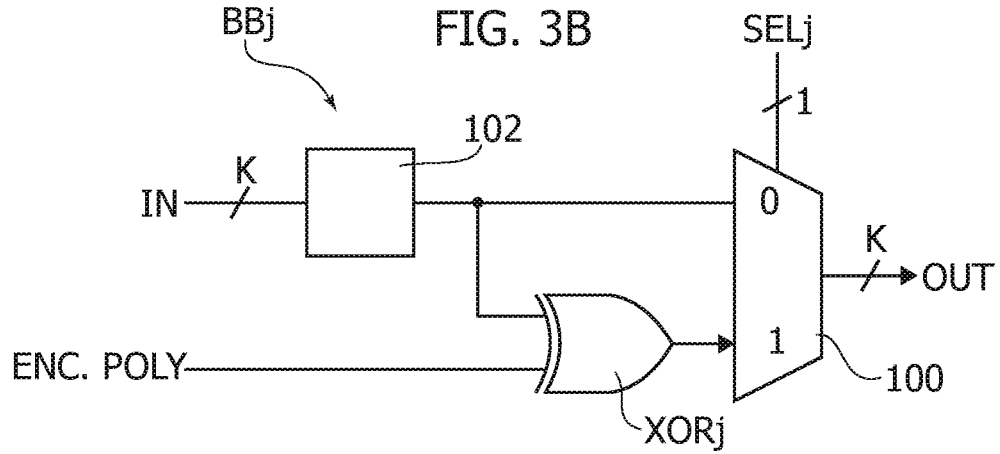

BBj
SELj
1
102
IN
K
0
K
OUT
1
ENC. POLY
100
XORj

FIG. 4

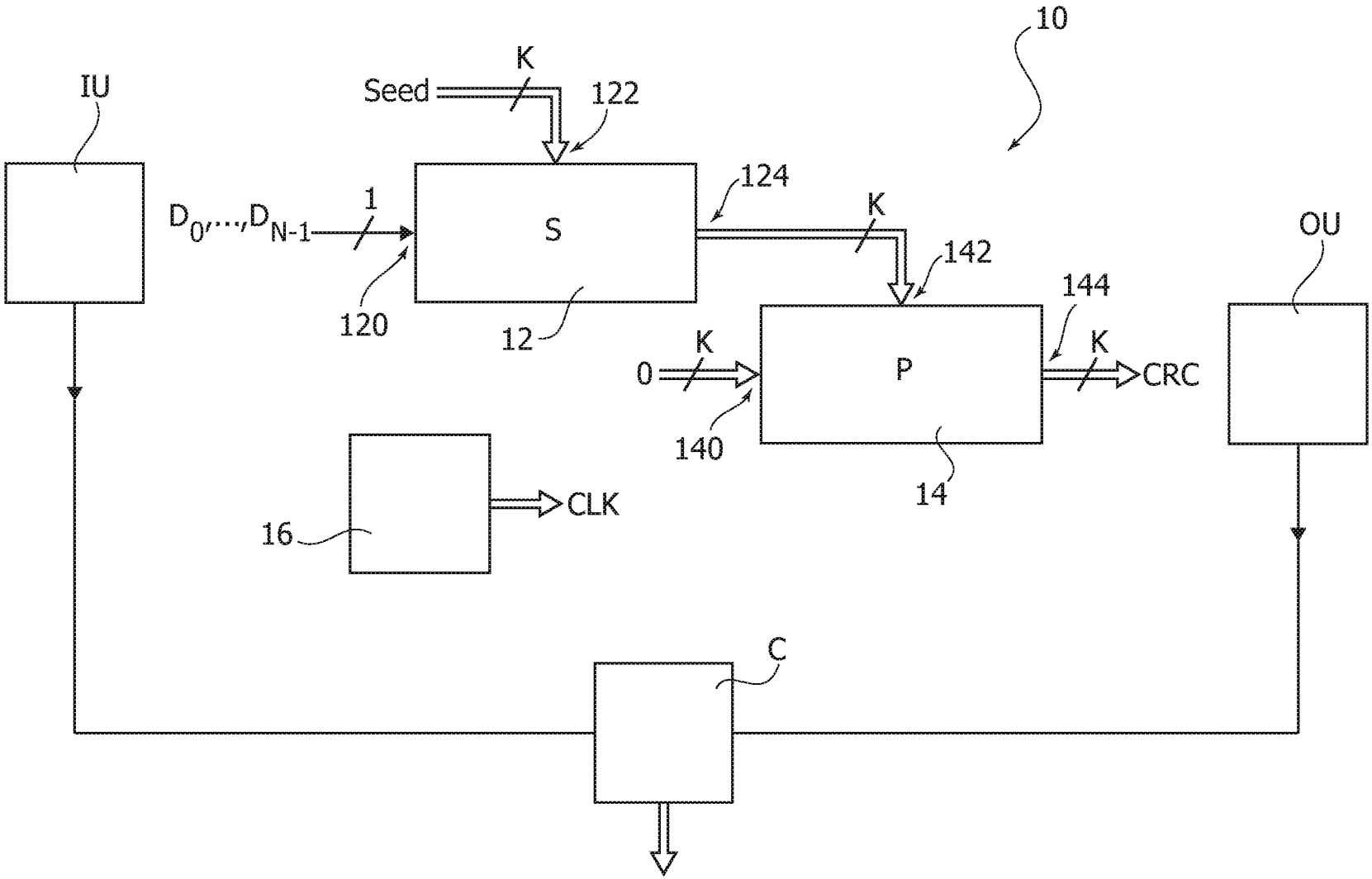
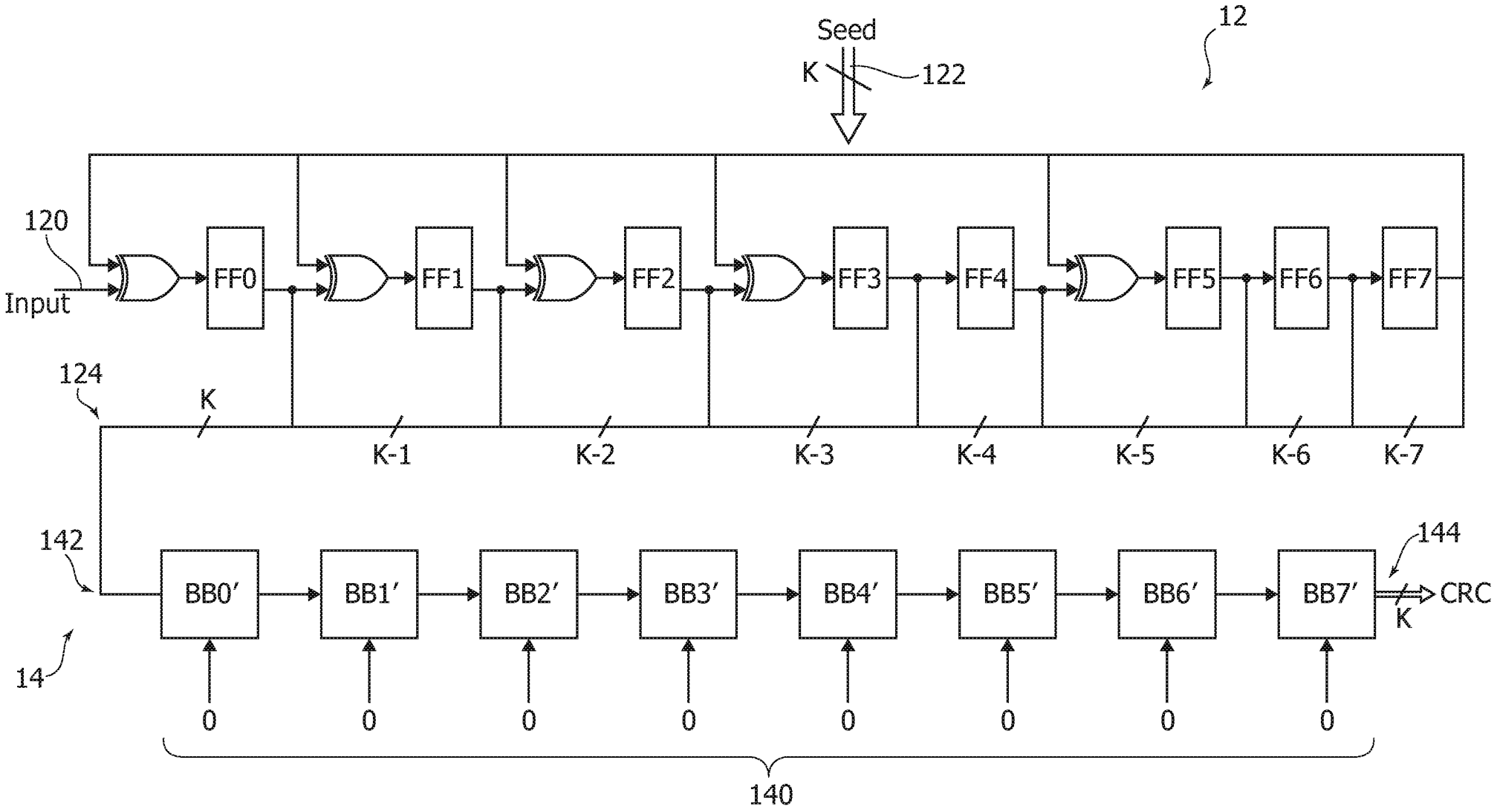

10
IU
Seed
K
122
$D_0,\ldots,D_{N-1}$
1
120
S
12
124
K
142
144
K
0
K
P
140
14
K
CRC
OU
16
CLK
C

FIG. 5


Seed
K
122
12
120
Input
FF0
FF1
FF2
FF3
FF4
FF5
FF6
FF7
124
K
K-1
K-2
K-3
K-4
K-5
K-6
K-7
142
144
BB0′
BB1′
BB2′
BB3′
BB4′
BB5′
BB6′
BB7′
CRC
K
14
0
0
0
0
0
0
0
0
140