

DOMANDA DI INVENZIONE NUMERO	102020000009358
Data Deposito	29/04/2020
Data Pubblicazione	29/10/2021

Classifiche IPC

Sezione	Classe	Sottoclasse	Gruppo	Sottogruppo
G	06	F	9	38

Titolo

Circuito, dispositivo, sistema e procedimento corrispondenti

DESCRIZIONE dell'invenzione industriale dal titolo:

"Circuito, dispositivo, sistema e procedimento corrispondenti"

di: STMicroelectronics S.r.l., di nazionalità italiana, via C. Olivetti, 2 - 20864 Agrate Brianza (MB)

Inventori designati: Giampiero BORGONOVO, Lorenzo RE FIORENTIN

Depositata il: 29 aprile 2020

TESTO DELLA DESCRIZIONE

Campo tecnico

La descrizione è relativa ai circuiti di elaborazione di segnale digitali, come gli acceleratori hardware, e a dispositivi e sistemi relativi.

Sfondo tecnologico

Una varietà di sistemi di elaborazione di segnale digitali in tempo reale ("real-time") (per es., per elaborare dati video e/o dati di immagini, dati radar, dati di comunicazioni senza fili ("wireless"), come sempre più richiesto nel campo automobilistico) possono comportare di elaborare una quantità rilevante di dati per unità di tempo. In varie applicazioni, tale elaborazione può risultare molto impegnativa per implementazioni puramente basate su core (cioè, implementazioni che coinvolgono microcontrollori o microprocessori non specializzati ("general purpose") che eseguono un software di elaborazione).

Perciò, l'uso di acceleratori hardware sta diventando sempre più rilevante in certi campi di elaborazione dati, siccome favorisce la velocizzazione del calcolo di certi

algoritmi. Acceleratori hardware progettati in modo appropriato possono ridurre il tempo di elaborazione di specifiche operazioni in confronto a implementazioni basate su core.

Gli acceleratori hardware tradizionali come descritti in letteratura o disponibili come prodotti commerciali possono comprendere differenti tipi di elementi di elaborazione (indicati anche come "unità matematiche" o "operatori matematici"), con ciascun elemento di elaborazione dedicato al calcolo di una specifica operazione. Per esempio, tali elementi di elaborazione possono comprendere circuiti di moltiplicazione ed accumulo (MAC, "multiply-and-accumulate") e/o circuiti configurati per calcolare funzioni di attivazione quali funzioni non lineari di attivazione (ANLF, "Activation Non-Linear Function") (per es., circuiti elaboratori digitali di rotazione di coordinate (CORDIC, "coordinate rotation digital computer")).

Ciascuno degli elementi di elaborazione summenzionati è progettato tipicamente per implementare una specifica funzione (per es., un algoritmo butterfly radice-2, una moltiplicazione di vettori complessi, un prodotto di vettori/matrici, una funzione trigonometrica o esponenziale o logaritmica, una convoluzione, ecc.). Perciò, un acceleratore hardware tradizionale comprende tipicamente una varietà di tali elementi di elaborazione differenti, connessi insieme attraverso qualche tipo di rete di interconnessione. In alcuni casi, i vari elementi di elaborazione sono attivati uno alla volta a causa delle dipendenze dei dati e/o delle limitazioni dell'architettura, conducendo a un utilizzo inefficiente dell'area di silicio e delle risorse hardware disponibili.

D'altra parte, un approccio basato su core, implementato puramente in software (per es., che fa uso di un processore di tipo SIMD, "Single Instruction Multiple Data") può comportare alte frequenze di clock al fine di soddisfare i tipici requisiti di larghezza di banda dei sistemi real-time, siccome in tal caso ciascun elemento di elaborazione esegue operazioni basilari.

Scopo e sintesi

Uno scopo di una o più forme di attuazione è fornire un dispositivo acceleratore hardware che risolve uno o più degli inconvenienti summenzionati.

In particolare, una o più forme di attuazione hanno come scopo fornire un dispositivo acceleratore hardware basato su memoria (indicato anche con l'acronimo EDPA, Enhanced Data Processing Architecture, nel contesto della presente descrizione) comprendente uno o più elementi di elaborazione. Gli elementi di elaborazione nel dispositivo acceleratore hardware possono essere riconfigurabili, eventualmente durante l'esecuzione (al run-time), così da fornire una maggiore flessibilità d'uso e facilitare un calcolo efficiente di una varietà di operazioni di elaborazione di segnale che possono essere particolarmente impegnative in termini delle risorse (per es., trasformata di Fourier veloce o FFT, filtraggio digitale, implementazione di reti neurali artificiali, ecc.).

Una o più forme di attuazione possono trovare applicazione in sistemi di elaborazione real-time in cui l'accelerazione delle operazioni impegnative dal punto di vista computazionale (per es., prodotto di vettori/matrici, convoluzione, FFT, algoritmo butterfly radice-2, moltiplicazione di vettori complessi, funzione

trigonometrica o esponenziale o logaritmica, ecc.) può favorire il soddisfacimento di certi requisiti di prestazioni (per es., in termini di tempo di elaborazione). Questo può essere il caso, per esempio, nel settore automobilistico.

Secondo una o più forme di attuazione, tale scopo può essere raggiunto per mezzo di un circuito (per es., un'unità di elaborazione riconfigurabile al run-time) avente le caratteristiche esposte nelle rivendicazioni che seguono.

Una o più forme di attuazione possono essere relative a un corrispondente dispositivo (per es., un circuito acceleratore hardware comprendente una o più unità di elaborazione riconfigurabili al run-time).

Una o più forme di attuazione possono essere relative a un corrispondente sistema (per es., un circuito integrato system-on-chip comprendente un circuito acceleratore hardware).

Una o più forme di attuazione possono essere relative a un corrispondente procedimento.

Le rivendicazioni sono parte integrante dell'insegnamento tecnico qui fornito con riferimento alle forme di attuazione.

Secondo una o più forme di attuazione, è fornito un circuito che può comprendere un insieme di terminali di ingresso configurati per ricevere segnali digitali di ingresso che trasportano dati di ingresso, e un insieme di terminali di uscita configurati per fornire segnali digitali di uscita che trasportano dati di uscita. Il circuito può comprendere circuiteria di calcolo configurata per produrre detti dati di uscita in funzione di detti dati di ingresso. La circuiteria di calcolo può comprendere un

insieme di circuiti moltiplicatori, un insieme di circuiti sommatori-sottrattori, un insieme di circuiti accumulatori e una rete di interconnessione configurabile. La rete di interconnessione configurabile può essere configurata per accoppiare selettivamente i circuiti moltiplicatori, i circuiti sommatori-sottrattori, i circuiti accumulatori, i terminali di ingresso e i terminali di uscita in almeno due configurazioni di elaborazione. In una prima configurazione di elaborazione, la circuiteria di calcolo può essere configurata per calcolare detti dati di uscita secondo un primo insieme di funzioni e, in almeno una seconda configurazione di elaborazione, la circuiteria di calcolo può essere configurata per calcolare detti dati di uscita secondo un rispettivo secondo insieme di funzioni. Il secondo insieme di funzioni è differente dal primo insieme di funzioni.

Una o più forme di attuazione possono pertanto fornire una maggiore flessibilità, un migliore utilizzo delle risorse hardware e/o migliori prestazioni per il calcolo parallelo.

Breve descrizione delle figure

Una o più forme di attuazione saranno ora descritte, a puro titolo di esempio, con riferimento alle figure annesse, nelle quali:

la Figura 1 è uno schema a blocchi circuitale esemplificativo di un sistema elettronico, come un system-on-chip, secondo una o più forme di attuazione,

la Figura 2 è uno schema a blocchi circuitale esemplificativo di un dispositivo elettronico che implementa un acceleratore hardware secondo una o più forme di attuazione,

la Figura 3 è uno schema a blocchi circuitale esemplificativo di un circuito di elaborazione secondo una o più forme di attuazione, per l'uso in un dispositivo elettronico secondo forme di attuazione,

la Figura 4 è un altro schema a blocchi circuitale esemplificativo di un circuito di elaborazione secondo una o più forme di attuazione, per l'uso in un dispositivo elettronico secondo forme di attuazione, e

la Figura 5 è un diagramma esemplificativo di una struttura di una rete di Percetroni Multistrato ("Multilayer Perceptron").

Descrizione dettagliata

Nella descrizione che segue, sono illustrati uno o più dettagli specifici, allo scopo di fornire una comprensione approfondita di esempi di forme di attuazione di questa descrizione. Le forme di attuazione possono essere ottenute senza uno o più dei dettagli specifici o con altri procedimenti, componenti, materiali, ecc. In altri casi, operazioni, materiali o strutture note non sono illustrate o descritte in dettaglio in modo tale che certi aspetti delle forme di attuazione non saranno resi poco chiari.

Un riferimento a "una forma di attuazione" nel quadro della presente descrizione intende indicare che una particolare configurazione, struttura, o caratteristica descritta con riferimento alla forma di attuazione è compresa in almeno una forma di attuazione. Per cui, le frasi come "in una forma di attuazione" o simili che possono essere presenti in uno o più punti della presente descrizione non fanno necessariamente riferimento proprio alla stessa forma di attuazione. Inoltre, particolari conformazioni, strutture o caratteristiche possono essere

combinare in un modo adeguato qualsiasi in una o più forme di attuazione.

In tutte le figure qui annesse, le parti o gli elementi simili sono indicati con riferimenti/numeri simili e una descrizione corrispondente non sarà ripetuta per brevità.

I riferimenti usati qui sono forniti semplicemente per convenienza e quindi non definiscono l'ambito di protezione o l'ambito delle forme di attuazione.

La Figura 1 è esemplificativa di un sistema elettronico 1 secondo una o più forme di attuazione, come un system-on-chip (SoC). Il sistema elettronico 1 può comprendere vari circuiti elettronici come, per esempio: un'unità di elaborazione centrale 10 ("Central Processing Unit"), per es., un microprocessore, una memoria di sistema principale 12 (per es., una memoria ad accesso casuale o RAM - Random Access Memory - di sistema), un controllore di accesso diretto alla memoria (DMA, "Direct Memory Access") 14 e un dispositivo acceleratore hardware 16.

Come esemplificato nella Figura 1, i circuiti elettronici nel sistema elettronico 1 possono essere connessi per mezzo di una rete di interconnessione di sistema 18, per es., un'interconnessione di SoC.

Uno scopo di una o più forme di attuazione è fornire un circuito acceleratore hardware 16 riconfigurabile (al run-time), progettato per supportare l'esecuzione di varie funzioni aritmetiche (basilari) con una maggiore flessibilità d'uso. Una o più forme di attuazione possono così favorire un miglioramento dell'utilizzo dell'area di silicio e fornire prestazioni di elaborazione soddisfacenti, per es., per soddisfare i requisiti di tempo

di elaborazione dei sistemi di elaborazione dati real-time.

Come esemplificato nella Figura 1, in una o più forme di attuazione un circuito acceleratore hardware 16 può comprendere almeno un elemento di elaborazione 160 configurabile (al run-time), preferibilmente un numero P di elementi di elaborazione $160_0, 160_1, \dots, 160_{P-1}$ configurabili (al run-time), e un insieme di banchi di memoria di dati locali M , preferibilmente un numero $Q=2*P$ di banchi di memoria di dati locali M_0, \dots, M_{Q-1} .

In una o più forme di attuazione, il circuito acceleratore hardware 16 può comprendere inoltre un'unità di controllo locale 161, una rete di interconnessione locale 162, un controllore della memoria di dati locale 163, un controllore della ROM locale 164 accoppiato a un insieme di memorie di sola lettura locali 165, preferibilmente un numero P di memorie di sola lettura locali $165_0, 165_1, \dots, 165_{P-1}$, e un controllore della memoria di configurazione locale 166 accoppiato a un insieme di memorie di coefficienti configurabili locali 167, preferibilmente un numero P di memorie di coefficienti configurabili locali $167_0, 167_1, \dots, 167_{P-1}$. Per esempio, le memorie 167 possono comprendere memorie volatili (per es., memorie RAM) e/o memorie non volatili (per es., memorie PCM).

Differenti forme di attuazione possono comprendere un numero P differente di elementi di elaborazione 160 e/o un numero Q differente di banchi di memoria di dati locali M_0, \dots, M_{Q-1} . A titolo di esempio, P può essere uguale a 8 e Q può essere uguale a 16.

In una o più forme di attuazione, gli elementi di elaborazione 160 possono essere configurabili in modo da supportare differenti funzioni di elaborazione (basilari)

con differenti livelli di parallelismo computazionale. Per esempio, un elemento di elaborazione 160 può supportare (per es., in base a una configurazione statica appropriata) differenti tipi di aritmetica (per es., 32 bit a singola precisione in virgola mobile, 32 bit interi/virgola fissa, oppure 16 o 8 bit con modalità vettorizzata o calcolo parallelo).

Gli elementi di elaborazione 160 possono comprendere rispettivi controllori di accesso diretto alla memoria (DMA) interni $168_0, 168_1, \dots, 168_{P-1}$ aventi una bassa complessità. In particolare, un elemento di elaborazione 160 può essere configurato per recuperare dati di ingresso dai banchi di memoria di dati locali M_0, \dots, M_{Q-1} e/o dalla memoria di sistema principale 12 attraverso il rispettivo controllore di accesso diretto alla memoria 168. L'elemento di elaborazione 160 può così elaborare i dati di ingresso recuperati per generare dati di uscita elaborati. L'elemento di elaborazione 160 può essere configurato per memorizzare i dati di uscita elaborati nei banchi di memoria di dati locali M_0, \dots, M_{Q-1} e/o nella memoria di sistema principale 12 attraverso il rispettivo controllore di accesso diretto alla memoria 168.

In aggiunta, gli elementi di elaborazione 160 possono essere configurati per recuperare dati di ingresso dalla memoria di sola lettura locale 165 e/o dalla memoria di coefficienti configurabili locale 167 per effettuare tale elaborazione.

In una o più forme di attuazione, il fatto di fornire un insieme di banchi di memoria di dati locali M_0, \dots, M_{Q-1} può facilitare l'elaborazione di dati in parallelo e la riduzione dei conflitti di accesso alla memoria.

Preferibilmente, i banchi di memoria di dati locali

M_0, \dots, M_{Q-1} possono essere provvisti di buffer (per es., doppio buffer), che può facilitare il recupero del tempo di caricamento (operazione di scrittura) e/o del tempo di scaricamento (operazione di lettura) della memoria. In particolare, ciascun banco di memoria di dati locale può essere duplicato, affinché i dati possano essere letti (per es., per essere elaborati) da uno dei due banchi di memoria e che (nuovi) dati possano essere memorizzati (per es., per essere elaborati successivamente) nell'altro banco di memoria allo stesso tempo. Come risultato, muovere i dati può non influenzare negativamente le prestazioni di calcolo, poiché può essere mascherato.

In una o più forme di attuazione, uno schema a doppio buffer dei banchi di memoria di dati locali M_0, \dots, M_{Q-1} può essere vantaggioso in combinazione con un'elaborazione dei dati in modalità streaming o back-to-back (per es., come applicabile in un processore FFT a N punti che è configurato per elaborare sequenze continue di N ingressi di dati).

In una o più forme di attuazione, i banchi di memoria di dati locali M_0, \dots, M_{Q-1} possono comprendere banchi di memoria aventi una capacità di memorizzazione limitata (e così, un'area di silicio limitata). Nel caso esemplificativo di un processore FFT, ciascuno dei banchi di memoria di dati locali può avere una capacità di memorizzazione di almeno $(\text{maxN})/Q$, dove maxN è la più lunga FFT che l'hardware è atto a trattare. Valori tipici in applicazioni che coinvolgono un acceleratore hardware possono essere i seguenti:

$N = 4096$ punti, con ciascun punto che è, per esempio, un numero complesso a singola precisione in virgola mobile (reale, img) avente una dimensione di 64 bit (o di 8 byte),

$P = 8$, avendo come risultato $Q = 16$, cosicché la capacità di memorizzazione di ciascun banco di memoria di dati locale può essere uguale a $(4096*8 \text{ byte})/16 = 2 \text{ KB}$ (KB = kilobyte).

In una o più forme di attuazione, l'unità di controllo locale 161 può comprendere un file di registro che comprende informazioni per impostare la configurazione degli elementi di elaborazione 160. Per esempio, l'unità di controllo locale 161 può impostare gli elementi di elaborazione 160 affinché effettuino uno specifico algoritmo come indicato da un'applicazione host in esecuzione sull'unità di elaborazione centrale 10.

In una o più forme di attuazione, l'unità di controllo locale 161 può così comprendere un circuito controllore del circuito acceleratore hardware 16. Un tale circuito controllore può configurare (per es., dinamicamente) ciascuno degli elementi di elaborazione 160 per calcolare una specifica funzione (basilare), e può configurare ciascuno dei rispettivi controllori di accesso diretto alla memoria interni 168 con uno specifico schema di accesso alla memoria e ciclo di loop.

In una o più forme di attuazione, la rete di interconnessione locale 162 può comprendere un sistema di interconnessione di bassa complessità, per es., basato su una rete a bus di un tipo noto, come un'interconnessione basata su AXI4. Per esempio, il parallelismo dei dati della rete di interconnessione locale 162 può essere su 64 bit e la larghezza dell'indirizzo può essere di 32 bit.

La rete di interconnessione locale 162 può essere configurata per connettere gli elementi di elaborazione 160 ai banchi di memoria di dati locali M_0, \dots, M_{Q-1} e/o alla memoria di sistema principale 12. In aggiunta, la rete di

interconnessione locale 162 può essere configurata per connettere l'unità di controllo locale 161 e il controllore della memoria di configurazione locale 166 alla rete di interconnessione di sistema 18.

In particolare, la rete di interconnessione 162 può comprendere:

un insieme di P porte master $MP_0, MP_1, \dots, MP_{P-1}$, ciascuna di queste porte master essendo accoppiata a un rispettivo elemento di elaborazione 160;

un insieme di P porte slave $SP_0, SP_1, \dots, SP_{P-1}$, ciascuna di queste porte slave essendo accoppiabile mediante il controllore della memoria di dati locale 163 ai banchi di memoria di dati locali M_0, \dots, M_{Q-1} ;

un'ulteriore coppia di porte comprendenti una porta master di sistema MP_P e una porta slave di sistema SP_P configurate per l'accoppiamento alla rete di interconnessione di sistema 18 (per es., al fine di ricevere istruzioni dall'unità di elaborazione centrale 10 e/o per accedere a dati memorizzati nella memoria di sistema 12); e

una porta slave SP_{P+1} ancora ulteriore accoppiata all'unità di controllo locale 161 e al controllore della memoria di configurazione locale 166.

In una o più forme di attuazione, la rete di interconnessione 162 può essere fissa (cioè, non riconfigurabile).

In un esempio di forma di attuazione (si veda, per es., la Tabella I-1 fornita in seguito, dove un segno "X" indica una connessione esistente tra due porte), la rete di interconnessione 162 può implementare le seguenti connessioni:

ciascuna delle P porte master $MP_0, MP_1, \dots, MP_{P-1}$

accoppiata a un elemento di elaborazione 160 può essere connessa a una rispettiva porta slave $SP_0, SP_1, \dots, SP_{P-1}$ accoppiata al controllore della memoria di dati locale 163; e

la porta master di sistema MP_p accoppiata alla rete di interconnessione di sistema 18 può essere connessa alla porta slave SP_{P+1} accoppiata all'unità di controllo locale 161 e al controllore della memoria di configurazione locale 166.

La Tabella I-1 fornita in seguito sintetizza tali esempi di connessioni implementate per mezzo della rete di interconnessione 162.

Tabella I-1

	SP_0	SP_1	...	SP_{P-1}	SP_p	SP_{P+1}
MP_0	X					
MP_1		X				
...			...			
MP_{P-1}				X		
MP_p						X

In un altro esempio di forma di attuazione (si veda, per es., la Tabella I-2 fornita in seguito), la rete di interconnessione 162 può implementare inoltre le seguenti connessioni: ciascuna delle P porte master $MP_0, MP_1, \dots, MP_{P-1}$ può essere connessa alla porta slave di sistema SP_p accoppiata alla rete di interconnessione di sistema 18. In questo modo, può essere fornita connettività tra un qualsiasi elemento di elaborazione 160 e il SoC mediante la rete di interconnessione di sistema 18.

La Tabella I-2 fornita in seguito sintetizza tali esempi di connessioni implementate per mezzo della rete di interconnessione 162.

Tabella I-2

	SP ₀	SP ₁	...	SP _{P-1}	SP _P	SP _{P+1}
MP ₀	X				X	
MP ₁		X			X	
...			
MP _{P-1}				X	X	
MP _P						X

In un altro esempio di forma di attuazione (si veda, per es., la Tabella I-3 fornita in seguito, dove un segno "X" indica una connessione esistente tra due porte, e una "X" tra parentesi indica una connessione opzionale), la rete di interconnessione 162 può implementare inoltre la seguente connessione: la porta master di sistema MP_P accoppiata alla rete di interconnessione di sistema 18 può essere connessa ad almeno una delle porte slave SP₀, SP₁, ..., SP_{P-1} (qui, la prima porta slave SP₀ nell'insieme di P porte slave SP₀, SP₁, ..., SP_{P-1}). In questo modo, può essere fornita una connessione tra la porta master MP_P e una (qualsiasi) porta slave. Secondo la specifica applicazione del sistema 1, una connessione della porta master MP_P può essere estesa a una pluralità delle (per es., a tutte le) porte slave SP₀, SP₁, ..., SP_{P-1}. Una connessione della porta master MP_P ad almeno una delle porte slave SP₀, SP₁, ..., SP_{P-1} può essere usata (soltanto) per caricare i dati di ingresso da elaborare nei banchi di memoria di dati locali M₀, ..., M_{Q-1}, nella misura in cui si può accedere a tutti i banchi di memoria mediante una singola porta slave. Il caricamento dei dati di ingresso può essere fatto usando soltanto una porta slave, mentre l'elaborazione di detti dati per mezzo di un calcolo parallelo può trarre vantaggio dal fatto di usare una pluralità delle (per es., tutte le) porte slave

$SP_0, SP_1, \dots, SP_{P-1}$.

La Tabella I-3 fornita in seguito sintetizza tali esempi di connessioni implementate per mezzo della rete di interconnessione 162.

Tabella I-3

	SP_0	SP_1	\dots	SP_{P-1}	SP_P	SP_{P+1}
MP_0	X				X	
MP_1		X			X	
\dots			\dots		\dots	
MP_{P-1}				X	X	
MP_P	X	(X)	(X)	(X)		X

In una o più forme di attuazione, il controllore della memoria di dati locale 163 può essere configurato per arbitrare l'accesso (per es., da parte degli elementi di elaborazione 160) ai banchi di memoria di dati locali M_0, \dots, M_{Q-1} . Per esempio, il controllore della memoria di dati locale 163 può usare schemi di accesso alla memoria selezionabili in funzione di un segnale ricevuto dall'unità di controllo locale 161 (per es., per il calcolo di uno specifico algoritmo).

In una o più forme di attuazione, il controllore della memoria di dati locale 163 può tradurre transazioni burst di lettura/scrittura di ingresso (per es., burst di tipo AXI) generate dai controllori di accesso diretto alla memoria in lettura/scrittura 168 in una sequenza di accessi alla memoria in lettura/scrittura, secondo un tipo di burst, una lunghezza del burst e uno schema di accesso alla memoria specificati.

Perciò, una o più forme di attuazione di un circuito acceleratore hardware 16 come esemplificato nella Figura 1 possono avere come scopo ridurre la complessità della rete di interconnessione locale 162 delegando al controllore

della memoria di dati locale 163 l'implementazione delle connessioni (riconfigurabili) tra gli elementi di elaborazione e i banchi di memoria di dati locali.

In una o più forme di attuazione, le memorie di sola lettura locali $165_0, 165_1, \dots, 165_{P-1}$ accessibili da parte degli elementi di elaborazione 160 mediante il controllore della ROM locale 164 possono essere configurate per memorizzare fattori numerici e/o coefficienti fissi per l'implementazione di specifici algoritmi o specifiche operazioni (per esempio, fattori di twiddle per il calcolo di FFT o altri coefficienti complessi). Il controllore della ROM locale 164 può implementare specifici schemi di indirizzi.

In una o più forme di attuazione, le memorie di coefficienti configurabili locali $167_0, 167_1, \dots, 167_{P-1}$ accessibili da parte degli elementi di elaborazione 160 mediante il controllore della memoria di configurazione locale 166 possono essere configurate per memorizzare coefficienti e/o fattori numerici dipendenti dall'applicazione che possono essere configurati mediante software (per esempio, coefficienti per implementare un filtro FIR o un'operazione di beamforming, pesi di una rete neurale, ecc.). Il controllore della memoria di configurazione locale 166 può implementare schemi di indirizzi specifici.

In una o più forme di attuazione, le memorie di sola lettura locali $165_0, 165_1, \dots, 165_{P-1}$ e/o le memorie di coefficienti configurabili locali $167_0, 167_1, \dots, 167_{P-1}$ possono essere suddivise vantaggiosamente in un numero P di banchi uguale al numero di elementi di elaborazione 160 compresi nel circuito acceleratore hardware 16. Ciò può facilitare il fatto di evitare conflitti durante calcoli

parallelisi. Per esempio, ciascuna delle memorie di coefficienti configurabili locali può essere configurata per fornire in parallelo l'insieme completo di coefficienti richiesti per ciascun elemento di elaborazione 160.

La Figura 2 è uno schema a blocchi circuitale esemplificativo di una o più forme di attuazione di un elemento di elaborazione 160 e delle relative connessioni al controllore della ROM locale 164, al controllore della memoria di configurazione locale 166 e ai banchi di memoria di dati locali M_0, \dots, M_{Q-1} (con le linee tratteggiate che indicano schematicamente una connessione riconfigurabile tra l'elemento di elaborazione 160 e i banchi di memoria di dati locali M_0, \dots, M_{Q-1} mediante la rete di interconnessione locale 162 e il controllore della memoria di dati locale 163).

Un elemento di elaborazione 160 come esemplificato nella Figura 2 può essere configurato per ricevere:

un primo segnale di ingresso P (per es., un segnale digitale indicativo di un valore binario da un banco di memoria di dati locale M_0, \dots, M_{Q-1} , eventualmente un dato complesso avente una parte reale e una parte immaginaria) mediante un rispettivo accesso diretto alla memoria in lettura 200₀ e un registro di buffer 202₀ (per es., un registro FIFO);

un secondo segnale di ingresso Q (per es., un segnale digitale indicativo di un valore binario da un banco di memoria di dati locale M_0, \dots, M_{Q-1} , eventualmente un dato complesso avente una parte reale e una parte immaginaria) mediante un rispettivo accesso diretto alla memoria in lettura 200₁ e un registro di buffer 202₁ (per es., un registro FIFO);

un primo coefficiente di ingresso W₀ (per es., un

segnaletico digitale indicativo di un valore binario da una memoria di sola lettura locale 165); e

un secondo, un terzo, un quarto e un quinto coefficiente di ingresso W_1, W_2, W_3, W_4 (per es., segnali digitali indicativi di rispettivi valori binari da una memoria di coefficienti configurabili locale 167).

In una o più forme di attuazione, un elemento di elaborazione 160 può comprendere un certo numero di accessi diretti alla memoria in lettura 200 uguale al numero di segnali di ingresso P, Q .

Si comprenderà che il numero di segnali di ingresso e/o di coefficienti di ingresso ricevuti nell'elemento di elaborazione 160 può variare in differenti forme di attuazione.

L'elemento di elaborazione 160 può comprendere un circuito di calcolo 20 configurabile (eventualmente al runtime) per elaborare i valori di ingresso P, Q e i coefficienti di ingresso W_0, W_1, W_2, W_3, W_4 per produrre un primo segnale di uscita X_0 (per es., un segnale digitale indicativo di un valore binario da memorizzare in un banco di memoria di dati locale M_0, \dots, M_{Q-1} mediante un rispettivo accesso diretto alla memoria in scrittura 204₀ e registro di buffer 206₀, quale un registro FIFO) e un secondo segnale di uscita X_1 (per es., un segnale digitale indicativo di un valore binario da memorizzare in un banco di memoria di dati locale M_0, \dots, M_{Q-1} mediante un rispettivo accesso diretto alla memoria in scrittura 204₁ e registro di buffer 206₁, come un registro FIFO).

In una o più forme di attuazione, un elemento di elaborazione 160 può comprendere un certo numero di accessi diretti alla memoria in scrittura 204 uguale al numero di segnali di uscita X_0, X_1 .

In una o più forme di attuazione, la programmazione degli accessi diretti alla memoria in lettura e/o scrittura 200, 204 (che sono compresi nei controllori di accesso diretto alla memoria 168) può essere effettuata mediante un'interfaccia (per es., un'interfaccia AMBA), che può consentire di accedere a registri di controllo interni situati nell'unità di controllo locale 161.

In aggiunta, un elemento di elaborazione 160 può comprendere un circuito generatore di indirizzi di ROM 208 accoppiato al controllore della ROM locale 164 e un circuito generatore di indirizzi di memoria 210 accoppiato al controllore della memoria di configurazione locale 166 per gestire il recupero dei dati da essi.

La Figura 3 è uno schema a blocchi circuitale esemplificativo di un circuito di calcolo 20 come compreso eventualmente in una o più forme di attuazione di un elemento di elaborazione 160.

Come esemplificato nella Figura 3, un circuito di calcolo 20 può comprendere un insieme di risorse di elaborazione comprendente, per esempio, quattro circuiti moltiplicatori complessi/reali (30a, 30b, 30c, 30d), due circuiti sommatori-sottrattori complessi (32a, 32b) e due circuiti accumulatori (34a, 34b) accoppiabili in modo riconfigurabile come illustrato sostanzialmente nella Figura 3. Per esempio, un accoppiamento riconfigurabile delle risorse di elaborazione per formare differenti percorsi di dati ("datapath"), con differenti datapath corrispondenti a differenti operazioni matematiche, può essere ottenuto per mezzo di circuiti multiplexer (per es., da 36a a 36j), con ciascun multiplexer che riceve un rispettivo segnale di controllo (per es., da S0 a S7).

In una o più forme di attuazione, i circuiti

moltiplicatori 30a, 30b, 30c, 30d possono essere configurati (per es., per mezzo di circuiti multiplexer interni non visibili nelle Figure) per funzionare secondo due configurazioni differenti, selezionabili in funzione di un segnale di controllo S8 fornito a essi. In una prima configurazione (per es., se $S8 = 0$), un moltiplicatore può calcolare il risultato di due prodotti reali su quattro operandi reali per ciclo di clock (cioè, ciascuno dei segnali di ingresso trasporta due valori che sono entrambi valori reali). In una seconda configurazione (per es., se $S8 = 1$), un moltiplicatore può calcolare il risultato di un prodotto complesso su due operandi complessi per ciclo di clock (cioè, ciascuno dei segnali di ingresso trasporta due valori, dove il primo valore è la parte reale dell'operando e il secondo valore è la parte immaginaria dell'operando).

La Tabella II fornita nel seguito riassume le possibili configurazioni esemplificative dei circuiti moltiplicatori 30a, 30b, 30c, 30d.

Tabella II

S8	Funzionamento dell'uscita		Ingressi
	Out_{re} (reale)	Out_{img} (immaginaria)	
0	$Out_{re}=In0_{re}*In1_{re}$	$Out_{img}=In0_{img}*In1_{img}$	$In0$: 2 dati reali
			$In1$: 2 dati reali
1	$Out_{re}=Re(Out_{img})$	$Out_{img}=Img(Out_{re})$	$In0$: 1 dati complessi
			$In1$: 1 data complessi

A titolo di esempio e con riferimento alla Figura 3, le risorse di elaborazione possono essere disposte come descritto nel seguito.

Un primo moltiplicatore 30a può ricevere un primo segnale di ingresso W1 e un secondo segnale di ingresso P (per es., operandi complessi).

Un secondo moltiplicatore 30b può ricevere un primo segnale di ingresso Q e un secondo segnale di ingresso selezionato tra i segnali di ingresso W2 e W4 per mezzo di un primo multiplexer 36a, il primo multiplexer 36a ricevendo un rispettivo segnale di controllo S2. Per esempio, se S2=0 allora il moltiplicatore 30b riceve il segnale W2 come secondo ingresso, e se S2=1 allora il moltiplicatore 30b riceve il segnale W4 come secondo ingresso.

Un terzo moltiplicatore 30c può ricevere un primo segnale di ingresso selezionato tra il segnale di uscita dal primo moltiplicatore 30a e il segnale di ingresso P.

Per esempio, come esemplificato nella Figura 3, un secondo multiplexer 36b può fornire come uscita, in funzione di un rispettivo segnale di controllo S0, il segnale di uscita dal primo moltiplicatore 30a (per es., se S0=0) o il segnale di ingresso P (per es., se S0=1). Un terzo multiplexer 36c può fornire come uscita al primo ingresso del terzo moltiplicatore 30c, in funzione di un rispettivo segnale di controllo S3, il segnale di uscita dal secondo multiplexer 36b (per es., se S3=1) o il segnale di ingresso P (per es., se S3=0).

Il terzo moltiplicatore 30c può ricevere un secondo segnale di ingresso selezionato tra il segnale di ingresso W3, il segnale di ingresso W4 e il segnale di ingresso W0.

Per esempio, come esemplificato nella Figura 3, un quarto multiplexer 36d può fornire come uscita, in funzione di un rispettivo segnale di controllo S3, il segnale di ingresso W4 (per es., se S3=0) o il segnale di ingresso W0 (per es., se S3=1). Un quinto multiplexer 36e può fornire come uscita al secondo ingresso del terzo moltiplicatore 30c, in funzione di un rispettivo segnale di controllo S3,

il segnale di ingresso W3 (per es., se S3=0) o il segnale di uscita dal quarto multiplexer 36d (per es., se S3=1).

Un quarto moltiplicatore 30d può ricevere un primo segnale di ingresso selezionato tra il segnale di ingresso Q e il segnale di uscita dal secondo moltiplicatore 30b.

Per esempio, come esemplificato nella Figura 3, un sesto multiplexer 36f può fornire come uscita al primo ingresso del quarto moltiplicatore 30d, in funzione di un rispettivo segnale di controllo S1, il segnale di ingresso Q (per es., se S1=0) o il segnale di uscita dal secondo moltiplicatore 30b (per es., se S1=1).

Il quarto moltiplicatore 30d può ricevere un secondo segnale di ingresso selezionato tra il segnale di ingresso W4 e il segnale di ingresso W0.

Per esempio, come esemplificato nella Figura 3, il secondo ingresso del quarto moltiplicatore 30d può essere accoppiato all'uscita del quarto multiplexer 36d.

Un primo sommatore-sottrattore 32a può ricevere un primo segnale di ingresso selezionato tra il segnale di uscita dal primo moltiplicatore 30a, il segnale di ingresso P e il segnale di uscita dal terzo moltiplicatore 30c.

Per esempio, come esemplificato nella Figura 3, un settimo multiplexer 36g può fornire come uscita al primo ingresso del primo sommatore-sottrattore 32a, in funzione di un rispettivo segnale di controllo S7, il segnale di uscita dal secondo multiplexer 36b (per es., se S7=1) o il segnale di uscita dal terzo moltiplicatore 30c (per es., se S7=0).

Il primo sommatore-sottrattore 32a può ricevere un secondo segnale di ingresso selezionato tra il segnale di ingresso Q, l'uscita dal secondo moltiplicatore 30b e un segnale nullo (cioè, un segnale binario uguale a zero).

Per esempio, come esemplificato nella Figura 3, un ottavo multiplexer 36h può fornire come uscita, in funzione di un rispettivo segnale di controllo S6, il segnale di ingresso Q (per es., se S6=0) o il segnale di uscita dal secondo moltiplicatore 30b (per es., se S6=1). Una prima porta logica AND 38a può ricevere come primo segnale di ingresso il segnale di uscita dall'ottavo multiplexer 36h, e come secondo segnale di ingresso un segnale di controllo G0. L'uscita della prima porta logica AND 38a può essere accoppiata al secondo ingresso del primo sommatore-sottrattore 32a.

Un secondo sommatore-sottrattore 32b può ricevere un primo segnale di ingresso selezionato tra il segnale di uscita dal terzo moltiplicatore 30c e il segnale di uscita dal quarto moltiplicatore 30d.

Per esempio, come esemplificato nella Figura 3, un nono multiplexer 36i può fornire come uscita al primo ingresso del secondo sommatore-sottrattore 32b, in funzione di un rispettivo segnale di controllo S5, il segnale di uscita dal terzo moltiplicatore 30c (per es., se S5=0) o il segnale di uscita dal quarto moltiplicatore 30d (per es., se S5=1).

Il secondo sommatore-sottrattore 32b può ricevere un secondo segnale di ingresso selezionato tra l'uscita dal quarto moltiplicatore 30d, l'uscita dal secondo moltiplicatore 30b e un segnale nullo (cioè, un segnale binario uguale a zero).

Per esempio, come esemplificato nella Figura 3, un decimo multiplexer 36j può fornire come uscita, in funzione di un rispettivo segnale di controllo S4, il segnale di uscita dal quarto moltiplicatore 30d (per es., se S4=0) o il segnale di uscita dal secondo moltiplicatore 30b (per

es., se $S4=1$). Una seconda porta logica AND 38b può ricevere come primo segnale di ingresso il segnale di uscita dal decimo multiplexer 36j, e come secondo segnale di ingresso un segnale di controllo G1. L'uscita della seconda porta logica AND 38b può essere accoppiata al secondo ingresso del secondo sommatore-sottrattore 32b.

Un primo accumulatore 34a può ricevere un segnale di ingresso dall'uscita del primo sommatore-sottrattore 32a, e un segnale di controllo EN, per fornire il primo segnale di uscita X0 del circuito di calcolo 20.

Un secondo accumulatore 34b può ricevere un segnale di ingresso dall'uscita del secondo sommatore-sottrattore 32b, e un segnale di controllo EN, per fornire il secondo segnale di uscita X1 del circuito di calcolo 20.

Una o più forme di attuazione comprendenti i sommatori-sottrattori 32a, 32b possono conservare la capacità di "bypassare" la loro operazione per mezzo delle porte logiche AND 38a, 38b che possono essere usate per forzare un segnale nullo al secondo ingresso dei sommatori-sottrattori 32a, 32b.

La Figura 4 è uno schema a blocchi circuitale esemplificativo di altre forme di attuazione di un circuito di calcolo 20 come compreso eventualmente in una o più forme di attuazione di un elemento di elaborazione 160.

Una o più forme di attuazione come esemplificate nella Figura 4 possono comprendere la stessa configurazione di risorse di elaborazione e di circuiti multiplexer come discussa con riferimento alla Figura 3, con l'aggiunta di due circuiti 40a, 40b configurati per calcolare funzioni non lineari di attivazione (ANLF), e rispettivi circuiti multiplexer 36k, 36m.

A titolo di esempio e con riferimento alla Figura 4,

le risorse di elaborazione 40a, 40b aggiuntive possono essere disposte come descritto nel seguito.

Un primo circuito di ANLF 40a può ricevere un segnale di ingresso dall'uscita del primo accumulatore 34a. Un undicesimo multiplexer 36k può fornire il primo segnale di uscita X0 del circuito di calcolo 20 selezionando, in funzione di un rispettivo segnale di controllo S9, il segnale di uscita dal primo accumulatore 34a (per es., se S9=0) o il segnale di uscita dal primo circuito di ANLF 40a (per es., se S9=1).

Un secondo circuito di ANLF 40b può ricevere un segnale di ingresso dall'uscita del secondo accumulatore 34b. Un dodicesimo multiplexer 36m può fornire il secondo segnale di uscita X1 del circuito di calcolo 20 selezionando, in funzione di un rispettivo segnale di controllo S9, il segnale di uscita dal secondo accumulatore 34b (per es., se S9=0) o il segnale di uscita dal secondo circuito di ANLF 40b (per es., se S9=1).

Perciò, in una o più forme di attuazione come esemplificate nella Figura 4, i circuiti di ANLF 40a e 40b possono essere "bypassati" per mezzo dei circuiti multiplexer 36k e 36m, fornendo con ciò un'operazione simile a quella delle forme di attuazione esemplificate nella Figura 3.

Perciò, come esemplificato con riferimento alla Figure 3 e 4, i datapath nel circuito di calcolo 20 possono essere configurati per supportare un calcolo parallelo e possono facilitare l'esecuzione di funzioni differenti. In una o più forme di attuazione, la pipeline interna può essere progettata per soddisfare i vincoli di temporizzazione (per es., la frequenza del clock) rispetto a una latenza minima.

In seguito, sono forniti vari esempi non limitativi di

possibili configurazioni del circuito di calcolo 20. In ciascun esempio, il circuito di calcolo 20 è configurato per calcolare una funzione (basilare) dipendente dall'algoritmo.

In un primo esempio, è descritta la configurazione di un circuito di calcolo 20 per effettuare un algoritmo di trasformata di Fourier veloce (FFT, "Fast Fourier Transform").

Nel caso in cui sia richiesto un circuito acceleratore hardware 16 per calcolare un algoritmo di FFT, un singolo elemento di elaborazione 160 può essere programmato per implementare un algoritmo butterfly di decimazione in frequenza o DIF ("Decimation In Frequency") radice-2 eseguendo le seguenti operazioni complesse, per es., usando segnali provenienti dall'unità di controllo interna 161:

$$X_0 = P+Q$$

$$X_1 = P*W_0 - Q*W_0$$

dove W_0 può essere un fattore di twiddle memorizzato all'interno di una memoria di sola lettura locale 165.

Nel presente primo esempio, i segnali di ingresso (P , Q , W_0 , W_1 , W_2 , W_3 , W_4) e i segnali di uscita (X_0 , X_1) possono essere del tipo dati complessi.

Opzionalmente, al fine di ridurre l'effetto sullo spettro delle discontinuità ai bordi dello spezzone di dati su cui è calcolato l'algoritmo di FFT, ai dati di ingresso può essere applicata una funzione di finestratura ("windowing") prima di calcolare l'algoritmo di FFT. Tale elaborazione di finestratura può essere supportata dall'elemento di elaborazione 160, per es., facendo uso di quattro circuiti moltiplicatori.

Opzionalmente, al posto del valore complesso può essere usato il modulo o la fase della componente dello

spettro (per es., in applicazioni come la rilevazione di un bersaglio tramite radar). In tal caso, i circuiti di ANLF interni (opzionali) possono essere usati durante l'ultimo stadio della FFT. Per esempio, è possibile ruotare il vettore complesso di ingresso in modo tale che sia allineato con l'asse x per calcolare il modulo.

La Tabella III fornita nel seguito riassume alcuni esempi di configurazioni del circuito di calcolo 20 per calcolare differenti algoritmi di radice-2.

Tabella III

Funzione	Configurazione												
	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	EN	G0	G1
Rdx-2 Butt.	1	0	-	1	0	0	0	1	1	0	0	1	1
Rdx-2 Butt. + Finestratura	0	1	0	1	0	0	1	1	1	0	0	1	1
Rdx-2 Butt. + Modulo	1	0	-	1	0	0	0	1	1	1	0	1	1

Perciò, il flusso di dati corrispondente alla funzione "Rdx-2 Butterfly" esemplificata sopra può essere il seguente:

$$X_0 = P+Q$$

$$X_1 = P*W_0 - Q*W_0$$

Il flusso di dati corrispondente alla funzione "Rdx-2 Butterfly + finestratura" esemplificata sopra può essere il seguente:

$$X_0 = W_1*P + W_2*Q$$

$$X_1 = (W_1*P)*W_0 - (W_2*Q)*W_0$$

Il flusso di dati corrispondente alla funzione "Rdx-2 Butterfly + modulo" esemplificata sopra può essere il seguente:

$$X_0 = \text{abs}(P+Q)$$

$$X_1 = \text{abs}(P*W_0 - Q*W_0)$$

Nel primo esempio qui considerato, la configurazione corrispondente alla funzione "Rdx-2 Butterfly" può comportare di usare due circuiti moltiplicatori, due circuiti sommatori-sottrattori, nessun accumulatore e nessun circuito di ANLF.

Nel primo esempio qui considerato, la configurazione corrispondente alla funzione "Rdx-2 Butterfly + finestratura" può comportare di usare quattro circuiti moltiplicatori, due circuiti sommatori-sottrattori, nessun accumulatore e nessun circuito di ANLF.

Nel primo esempio qui considerato, la configurazione corrispondente alla funzione "Rdx-2 Butterfly + modulo" può comportare di usare due circuiti moltiplicatori, due circuiti sommatori-sottrattori, nessun accumulatore e due circuiti di ANLF.

In un secondo esempio, è descritta la configurazione di un circuito di calcolo 20 per eseguire un prodotto scalare di vettori di dati complessi.

Per calcolare un prodotto scalare di vettori di dati complessi può essere richiesto un circuito acceleratore hardware 16. Per esempio, questo può essere il caso di applicazioni che comportano un'operazione di filtraggio, come i sistemi radar "phased-array" che comportano uno stadio di elaborazione detto "beamforming". Il beamforming può facilitare a un sistema radar di risolvere i bersagli anche nella dimensione angolare (azimuth), oltre a quelle della distanza e della velocità radiale.

Nel presente secondo esempio, i segnali di ingresso (P , Q , W_0 , W_1 , W_2 , W_3 , W_4) e i segnali di uscita (X_0 , X_1) possono essere del tipo dati complessi.

Nel presente secondo esempio, due differenti operazioni di prodotto scalare di vettori (per es.,

operazioni di beamforming) possono essere realizzate simultaneamente da un singolo elemento di elaborazione 160, per es., facendo uso di tutte le risorse hardware interne.

Durante le operazioni di beamforming, la memoria di coefficienti configurabili locale 167 può essere usata per memorizzare gli sfasamenti per i differenti elementi di antenna della schiera.

In modo simile al primo esempio, anche nel presente secondo esempio può essere usato opzionalmente un circuito di ANLF se è da calcolare un modulo invece di un valore complesso.

La Tabella IV fornita nel seguito esemplifica possibili configurazioni del circuito di calcolo 20 per calcolare simultaneamente due prodotti scalari di vettori.

Tabella IV

Funzione	Configurazione												
	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	EN	G0	G1
2x prodotto scalare di vettori (complessi)	0	0	0	0	0	0	1	1	1	0	1	1	1
2x prodotto scalare di vettori (complessi) + Modulo	0	0	0	0	0	0	1	1	1	1	1	1	1

Perciò, il flusso di dati corrispondente alla funzione "prodotto scalare di vettori" sopra esemplificata può essere il seguente:

$$X0 = ACC(P*W1 + Q*W2)$$

$$X1 = ACC(P*W3 + Q*W4)$$

Il flusso di dati corrispondente alla funzione "prodotto scalare di vettori + modulo" sopra esemplificata può essere il seguente:

```
X0 = abs(ACC(P*W1 + Q*W2))
X1 = abs(ACC(P*W3 + Q*W4))
```

Nel secondo esempio qui considerato, la configurazione corrispondente alla funzione "prodotto scalare di vettori" può comportare l'uso di quattro circuiti moltiplicatori, due circuiti sommatori-sottrattori, due accumulatori e nessun circuito di ANLF.

Nel secondo esempio qui considerato, la configurazione corrispondente alla funzione "prodotto scalare di vettori + modulo" può comportare l'uso di quattro circuiti moltiplicatori, due circuiti sommatori-sottrattori, due accumulatori e due circuiti di ANLF.

In un terzo esempio, è descritta la configurazione di un circuito di calcolo 20 per eseguire un prodotto scalare di vettori di dati reali.

Per calcolare un prodotto scalare di vettori di dati reali su grandi strutture di dati reali, per es., per il calcolo di un filtro digitale può essere necessario un circuito acceleratore hardware 16. Per esempio, in molte applicazioni, un segnale del mondo reale (per es., analogico) può essere sottoposto a filtraggio dopo essere stato reso digitale, al fine di estrarre (soltanto) le informazioni rilevanti.

Nel dominio digitale, un'operazione di convoluzione tra un segnale di ingresso e una risposta all'impulso di un filtro (FIR, "Filter Impulse Response") può assumere la forma di un prodotto scalare di due vettori di dati reali. Uno dei vettori può contenere i dati di ingresso, mentre l'altro può contenere i coefficienti che definiscono l'operazione di filtraggio.

Nel presente terzo esempio, i segnali di ingresso (P , Q , W_0 , W_1 , W_2 , W_3 , W_4) e i segnali di uscita (X_0 , X_1) sono

del tipo dati reali.

Nel presente terzo esempio, possono essere realizzate simultaneamente due operazioni di filtraggio differenti da parte di un singolo elemento di elaborazione 160 sullo stesso insieme di dati, elaborando quattro dati di ingresso differenti per ciclo di clock, per es., facendo uso di tutte le risorse hardware interne.

La Tabella V fornita nel seguito rappresenta a titolo di esempio una possibile configurazione del circuito di calcolo 20 per calcolare simultaneamente due operazioni di filtraggio su vettori di dati reali.

Tabella V

Funzione	Configurazione												
	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	EN	G0	G1
2x prodotto scalare di vettori (reali)	0	0	0	0	0	0	1	1	0	0	1	1	1

Perciò, il flusso di dati corrispondente alle funzioni sopra esemplificate è il seguente, dove il pedice "h" sta per la parte di MSB, e il pedice "l" sta per la parte di LSB:

$$X0_h = ACC(P_h * W1_h + Q_h * W2_h)$$

$$X0_l = ACC(P_l * W1_l + Q_l * W2_l)$$

$$X1_h = ACC(P_h * W3_h + Q_h * W4_h)$$

$$X1_l = ACC(P_l * W3_l + Q_l * W4_l)$$

Nel terzo esempio qui considerato, la configurazione corrispondente alla funzione "prodotto scalare di vettori reali" può comportare l'uso di quattro circuiti moltiplicatori, due circuiti sommatori-sottrattori, due accumulatori e nessun circuito di ANLF.

In un quarto esempio, è descritta la configurazione di un circuito di calcolo 20 per calcolare funzioni non

lineari.

Il Percettrone Multistrato (MLP, "MultiLayer Perceptron") è una classe di reti neurali artificiali feedforward completamente connesse, che possono comprendere almeno tre strati di nodi/neuroni. Ciascun neurone, ad eccezione di quelli nello strato di ingresso, calcola una somma ponderata di tutti i nodi nello strato precedente e applica poi una funzione di attivazione non lineare al risultato. Un elemento di elaborazione 160 come qui descritto può trattare tali funzioni non lineari, per es., usando i circuiti di ANLF interni. Tipicamente, le reti neurali si occupano di dati provenienti dal mondo reale e usano pesi reali e funzioni per calcolare le probabilità di appartenenza a una classe (le uscite dell'ultimo strato). Prodotti scalari di dati reali possono così essere le operazioni usate più frequentemente e con le maggiori esigenze dal punto di vista computazionale per tale genere di reti artificiali.

La Figura 5 è un diagramma esemplificativo di una tipica struttura di una rete di Percettroni Multistrato 50.

Come esemplificato nella Figura 5, una rete di Percettroni Multistrato 50 può comprendere:

uno strato di ingresso 50a comprendente un numero N di ingressi U^1, \dots, U^N ($U^i, i = 1, \dots, N$),

uno strato nascosto 50b comprendente un numero M di nodi nascosti X^1, \dots, X^M ($X^k, k = 1, \dots, M$), e

uno strato di uscita 50c comprendente un numero P di nodi di uscita Y^1, \dots, Y^P ($Y^j, j = 1, \dots, P$).

Si comprenderà che, in una o più forme di attuazione, una rete di Percettroni Multistrato può comprendere più di uno strato nascosto 50b.

Come esemplificato nella Figura 5, una rete di

Percettroni Multistrato 50 può comprendere un primo insieme di $N \times M$ pesi $W^{i,k}$ tra gli ingressi U^1, \dots, U^N e i nodi nascosti X^1, \dots, X^M , e un secondo insieme di $M \times P$ pesi $W^{k,j}$ tra i nodi nascosti X^1, \dots, X^M e i nodi di uscita Y^1, \dots, Y^P .

I valori memorizzati negli ingressi U^i , nei nodi nascosti X^k e nei nodi di uscita Y^j possono essere calcolati, per esempio, come virgole mobili MAC con singola precisione.

I valori dei nodi nascosti X^k e dei nodi di uscita Y^j possono essere calcolati secondo le seguenti equazioni:

$$X^k = f \left(\sum_{i=1}^N (U^i W^{i,k}) \right)$$

$$Y^j = f \left(\sum_{i=1}^M (X^i W^{k,j}) \right)$$

Nel presente quarto esempio, i pesi reali "addestrati" associati a tutte le connessioni di un MLP possono essere memorizzati all'interno della memoria di coefficienti configurabili locale 167. Gli ingressi dello strato reale possono essere recuperati dalla memoria di dati locale del circuito acceleratore hardware 16 (per es., i banchi di memoria di dati locali M_0, \dots, M_{Q-1}) e le uscite dello strato reale possono essere memorizzate nella memoria di dati locale del circuito acceleratore hardware 16.

Come risultato di un modello di MLP che è mappato sul circuito acceleratore hardware 16, ciascuno degli elementi di elaborazione 160 inclusi al suo interno (per es., un numero P di elementi di elaborazione) può essere usato per calcolare il prodotto scalare e l'uscita della funzione di attivazione associata a due neuroni differenti dello stesso strato, per es., trattando quattro archi per ciclo di clock. Come risultato, possono essere usati simultaneamente

tutti gli elementi di elaborazione $160_0, 160_1, \dots, 160_{P-1}$.

La Tabella VI fornita nel seguito rappresenta a titolo di esempio una possibile configurazione del circuito di calcolo 20 per calcolare simultaneamente due uscite di funzione di attivazione associate a due neuroni differenti.

Tabella VI

Funzione	Configurazione												
	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	EN	G0	G1
Motore di calcolo di MLP	0	0	0	0	0	0	1	1	0	1	1	1	1

Perciò, il flusso di dati corrispondente alle funzioni esemplificate in precedenza è il seguente, dove il pedice "h" sta per la parte di MSB, e il pedice "l" sta per la parte di LSB:

$$X0_h = f(ACC(P_h * W1_h + Q_h * W2_h))$$

$$X0_l = f(ACC(P_l * W1_l + Q_l * W2_l))$$

$$X1_h = f(ACC(P_h * W3_h + Q_h * W4_h))$$

$$X1_l = f(ACC(P_l * W3_l + Q_l * W4_l))$$

Nel quarto esempio qui considerato, la configurazione corrispondente alla funzione "motore di calcolo di MLP" (che può comprendere calcolare due prodotti scalari di vettori e applicare a essi una funzione di attivazione non lineare) può comportare l'uso di quattro circuiti moltiplicatori, due circuiti sommatori-sottrattori, due accumulatori e due circuiti di ANLF.

La Tabella VII fornita nel seguito rappresenta a titolo di esempio funzioni non lineari che possono essere implementate in una o più forme di attuazione. Alcune funzioni indicate con "Algoritmo = NN" possono essere usate specificamente nel contesto delle reti neurali.

Tabella VII

Nome	Funzione	Algoritmo
Sigmoide (σ) Funzione logistica	$f(x) = \frac{1}{1 + e^{-x}}$	NN
tanh tangente iperbolica	$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	NN
ReLU Unità Lineare Raddrizzata	$f(x) = \begin{cases} 0 & \text{per } x < 0 \\ x & \text{per } x \geq 0 \end{cases}$	NN
Softmax	$f(x)_i = \frac{e^{x_i}}{\sum_k e^{x_k}}$	NN
Leaky_ReLU	$f(x) = \begin{cases} \alpha x & \text{per } x < 0 \\ x & \text{per } x \geq 0 \end{cases}$	NN
SELU ELU Scalata	$f(x) = \lambda \cdot \begin{cases} \alpha(e^x - 1) & \text{per } x < 0 \\ x & \text{per } x \geq 0 \end{cases}$	NN
Sigmoide hard	$f(x) = \begin{cases} 0 & \text{per } x < 2.5 \\ 0.2x + 0.5 & \text{per } -2.5 \leq x \leq 2.5 \\ 1 & \text{per } x > 2.5 \end{cases}$	NN
ELU LU Esponenziale	$f(x) = \begin{cases} (e^x - 1) & \text{per } x < 0 \\ x & \text{per } x \geq 0 \end{cases}$	NN
Log (x)	$f(x) = \log(x) \text{ per } x \geq 0$	
Sin(x)	$f(x) = \sin(x)$	
Cos(x)	$f(x) = \cos(x)$	
Abs(x)	$f(x) = \begin{cases} -x & \text{per } x < 0 \\ x & \text{per } x \geq 0 \end{cases}$	
Sqrt(x)	$f(x) = \sqrt{x} \text{ for } x \geq 0$	
Modulo ($z = re + im$) Fase ($z = re + im$)	$m(z) = \sqrt{(re^2 + im^2)}$ $ph(z) = \arctan \frac{im}{re}$	

Perciò, una o più forme di attuazione di un circuito acceleratore hardware 16 comprendente almeno un circuito di calcolo 20 come qui esemplificato e/o negli esempi precedenti possono facilitare l'implementazione di un sistema di elaborazione di segnale digitale con uno o più dei vantaggi seguenti:

flessibilità (per es., capacità di trattare differenti tipi di algoritmi),

megliore utilizzo delle risorse hardware,
migliori prestazioni per il calcolo parallelo,
connettività estesa e larghezza di banda elevata di
ciascun elemento di elaborazione 160 ai banchi di memoria
di dati locali M_0, \dots, M_{Q-1} e/o alla memoria di sistema 12
attraverso una semplice rete di interconnessione locale 162
e controllori di accesso diretto alla memoria $168_0, 168_1,$
 $\dots, 168_{P-1}$ interni, e

architettura scalabile per supportare algoritmi
aggiuntivi con l'integrazione di differenti elementi di
elaborazione.

In una o più forme di attuazione, il sistema
elettronico 1 può essere implementato come un circuito
integrato in un singolo chip o die di silicio (per es.,
come un system-on-chip). In alternativa, il sistema
elettronico 1 può essere un sistema distribuito
comprendente una pluralità di circuiti integrati
interconnessi insieme, per es., per mezzo di una scheda a
circuiti stampati (PCB, "Printed Circuit Board").

Come qui esemplificato, un circuito (per es., 160) può
comprendere un insieme di terminali di ingresso configurati
per ricevere un rispettivo insieme di segnali digitali di
ingresso che trasportano dati di ingresso (per es., $P, Q,$
 W_0, W_1, W_2, W_3, W_4), un insieme di terminali di uscita
configurati per fornire un rispettivo insieme di segnali
digitali di uscita che trasportano dati di uscita (per es.,
 X_0, X_1), e una circuiteria di calcolo (per es., 20)
configurata per produrre detti dati di uscita in funzione
di detti dati di ingresso. La circuiteria di calcolo può
comprendere:

un insieme di circuiti moltiplicatori (per es., 30a,
30b, 30c, 30d),

un insieme di circuiti sommatori-sottrattori (per es., 32a, 32b),

un insieme di circuiti accumulatori (per es., 34a, 34b), e

una rete di interconnessione configurabile (per es., 36a, ..., 36j) configurata per accoppiare selettivamente (per es., S₁, ..., S₇) detti circuiti moltiplicatori, detti circuiti sommatori-sottrattori, detti circuiti accumulatori, detti terminali di ingresso e detti terminali di uscita in almeno due configurazioni di elaborazione.

Come qui esemplificato, in una prima configurazione di elaborazione detta circuiteria di calcolo può essere configurata per calcolare detti dati di uscita secondo un primo insieme di funzioni, e in almeno una seconda configurazione di elaborazione detta circuiteria di calcolo può essere configurata per calcolare detti dati di uscita secondo un rispettivo secondo insieme di funzioni, detto rispettivo secondo insieme di funzioni essendo differente da detto primo insieme di funzioni.

Come qui esemplificato, un circuito può comprendere rispettivi controllori di accesso diretto alla memoria in lettura configurabili (per es., 200₀, 200₁) accoppiati a un primo sottoinsieme di detto insieme di terminali di ingresso per ricevere (per es., 162, 163) un rispettivo primo sottoinsieme di detti segnali digitali di ingresso che trasportano un primo sottoinsieme di dati di ingresso (per es., P, Q). I controllori di accesso diretto alla memoria in lettura configurabili possono essere configurati per controllare il prelevamento di detto primo sottoinsieme di dati di ingresso da una memoria (per es., M₀, ..., M_{Q-1}).

Come qui esemplificato, un circuito può comprendere rispettivi controllori di accesso diretto alla memoria in

scrittura configurabili (per es., 204₀, 204₁) accoppiati a detto insieme di terminali di uscita per fornire detti segnali digitali di uscita che trasportano dati di uscita. I controllori di accesso diretto alla memoria in scrittura configurabili possono essere configurati per controllare la memorizzazione di detti dati di uscita in detta memoria.

Come qui esemplificato, un circuito può comprendere rispettivi registri di buffer di ingresso (per es., 202₀, 202₁) accoppiati a detti controllori di accesso diretto alla memoria in lettura configurabili e rispettivi registri di buffer di uscita (per es., 206₀, 206₁) accoppiati a detti controllori di accesso diretto alla memoria in scrittura configurabili.

Come qui esemplificato, un circuito può comprendere un circuito generatore di indirizzi di ROM (per es., 208) configurato per controllare il prelevamento di un secondo sottoinsieme di dati di ingresso (per es., W0) da almeno una memoria di sola lettura (per es., 164, 165) mediante un secondo sottoinsieme di detti segnali digitali di ingresso e/o un circuito generatore di indirizzi di memoria (per es., 210) configurato per controllare il prelevamento di un terzo sottoinsieme di dati di ingresso (per es., W1, W2, W3, W4) da almeno una memoria configurabile (per es., 166, 167) mediante un terzo sottoinsieme di detti segnali digitali di ingresso.

Come qui esemplificato, in un circuito secondo forme di attuazione detto insieme di circuiti moltiplicatori può comprendere un primo circuito moltiplicatore (per es., 30a), un secondo circuito moltiplicatore (per es., 30b), un terzo circuito moltiplicatore (per es., 30c) e un quarto circuito moltiplicatore (per es., 30d). L'insieme di circuiti sommatori-sottrattori può comprendere un primo

circuito sommatore-sottrattore (per es., 32a) e un secondo circuito sommatore-sottrattore (32b). L'insieme di circuiti accumulatori può comprendere un primo circuito accumulatore (per es., 34a) e un secondo circuito accumulatore (per es., 34b).

Come qui esemplificato, il primo circuito moltiplicatore può ricevere come primo operando un primo segnale di ingresso (per es., W1) di detto insieme di segnali digitali di ingresso, e può ricevere come secondo operando un secondo segnale di ingresso (per es., P) di detto insieme di segnali digitali di ingresso. Il secondo circuito moltiplicatore può ricevere come primo operando un terzo segnale di ingresso (per es., Q) di detto insieme di segnali digitali di ingresso, e può ricevere come secondo operando un segnale selezionabile tra un quarto segnale di ingresso (per es., W2) e un quinto segnale di ingresso (per es., W4) di detto insieme di segnali digitali di ingresso. Il terzo circuito moltiplicatore può ricevere come primo operando un segnale selezionabile tra un segnale di uscita da detto primo circuito moltiplicatore e detto secondo segnale di ingresso, e può ricevere come secondo operando un segnale selezionabile tra un sesto segnale di ingresso (per es., W3), un settimo segnale di ingresso (per es., W0) e detto quinto segnale di ingresso. Il quarto circuito moltiplicatore può ricevere come primo operando un segnale selezionabile tra un segnale di uscita da detto secondo moltiplicatore e detto terzo segnale di ingresso, e può ricevere come secondo operando un segnale selezionabile tra detto quinto segnale di ingresso e detto settimo segnale di ingresso. Il primo circuito sommatore-sottrattore può ricevere come primo operando un segnale selezionabile tra detto segnale di uscita da detto primo circuito

moltiplicatore, detto secondo segnale di ingresso e detto segnale di uscita da detto terzo circuito moltiplicatore, e può ricevere come secondo operando un segnale selezionabile tra detto terzo segnale di ingresso, detto segnale di uscita da detto secondo circuito moltiplicatore e un segnale nullo. Il secondo circuito sommatore-sottrattore può ricevere come primo operando un segnale selezionabile tra detto segnale di uscita da detto terzo circuito moltiplicatore e detto segnale di uscita da detto quarto circuito moltiplicatore, e può ricevere come secondo operando un segnale selezionabile tra detto segnale di uscita da detto quarto circuito moltiplicatore, detto segnale di uscita da detto secondo circuito moltiplicatore e un segnale nullo. Il primo circuito accumulatore può ricevere come ingresso un segnale di uscita da detto primo circuito sommatore-sottrattore, e il secondo circuito accumulatore può ricevere come ingresso un segnale di uscita da detto secondo circuito sommatore-sottrattore. Il primo circuito accumulatore può essere selettivamente attivabile (per es., EN) per fornire un primo segnale di uscita (per es., X0) e detto secondo circuito accumulatore può essere selettivamente attivabile per fornire un secondo segnale di uscita (per es., X1).

Come qui esemplificato, detta circuiteria di calcolo può comprendere un insieme di circuiti configurati per calcolare funzioni non lineari (per es., 40a, 40b).

Come qui esemplificato, detto insieme di circuiti configurati per calcolare funzioni non lineari può comprendere un primo circuito configurato per calcolare funzioni non lineari (per es., 40a) e un secondo circuito configurato per calcolare funzioni non lineari (per es., 40b). Il primo circuito configurato per calcolare funzioni

non lineari può ricevere come ingresso un segnale di uscita da detto primo circuito accumulatore. Il secondo circuito configurato per calcolare funzioni non lineari può ricevere come ingresso un segnale di uscita da detto secondo circuito accumulatore. Il primo segnale di uscita può essere selezionabile (per es., 36k) tra il segnale di uscita dal primo circuito accumulatore e il segnale di uscita dal primo circuito configurato per calcolare funzioni non lineari, e il secondo segnale di uscita può essere selezionabile (per es., 36m) tra il segnale di uscita dal secondo circuito accumulatore e il segnale di uscita dal secondo circuito configurato per calcolare funzioni non lineari.

Come qui esemplificato, un dispositivo (per es., 16) può comprendere un insieme di circuiti secondo una o più forme di attuazione, un insieme di banchi di memoria di dati (per es., M_0, \dots, M_{Q-1}) e un'unità di controllo (per es., 161). I circuiti possono essere configurabili (per es., 161, 168) per leggere dati da e scrivere dati in detti banchi di memoria di dati mediante una rete di interconnessione (per es., 162, 163) in funzione di dati di configurazione memorizzati in detta unità di controllo.

Come qui esemplificato, detti banchi di memoria di dati possono comprendere registri di buffer, preferibilmente registri di buffer doppi.

Come qui esemplificato, un sistema (per es., 1) può comprendere un dispositivo secondo una o più forme di attuazione e un'unità di elaborazione (per es., 10) accoppiata al dispositivo mediante un'interconnessione di sistema (per es., 18). I circuiti nell'insieme di circuiti del dispositivo possono essere configurabili in almeno due configurazioni di elaborazione in funzione di un segnale di

controllo ricevuto dall'unità di elaborazione.

Come qui esemplificato, un procedimento di funzionamento di un circuito secondo una o più forme di attuazione, un dispositivo secondo una o più forme di attuazione, o un sistema secondo una o più forme di attuazione può comprendere partizionare il tempo di funzionamento di detta circuiteria di calcolo in almeno un primo e un secondo intervallo di funzionamento, in cui detta circuiteria di calcolo è fatta funzionare rispettivamente in detta prima configurazione di elaborazione e in detta almeno una seconda configurazione di elaborazione.

Fermi restando i principi di fondo, i dettagli e le forme di attuazione possono variare, anche in modo apprezzabile, rispetto a quanto è stato descritto, puramente a titolo di esempio, senza uscire dall'ambito di protezione.

L'ambito di protezione è definito dalle rivendicazioni annesse.

RIVENDICAZIONI

1. Un circuito (160) comprendente:

un insieme di terminali di ingresso configurati per ricevere un insieme di rispettivi segnali digitali di ingresso che trasportano dati di ingresso (P, Q, W0, W1, W2, W3, W4),

un insieme di terminali di uscita configurati per fornire un insieme di rispettivi segnali digitali di uscita che trasportano dati di uscita (X0, X1), e

circuiteria di calcolo (20) configurata per produrre detti dati di uscita in funzione di detti dati di ingresso, in cui la circuiteria di calcolo (20) comprende:

un insieme di circuiti moltiplicatori (30a, 30b, 30c, 30d),

un insieme di circuiti sommatori-sottrattori (32a, 32b),

un insieme di circuiti accumulatori (34a, 34b), e

una rete di interconnessione configurabile (36a, ..., 36j) configurata per accoppiare selettivamente (S1, ..., S7) detti circuiti moltiplicatori (30a, 30b, 30c, 30d), detti circuiti sommatori-sottrattori (32a, 32b), detti circuiti accumulatori (34a, 34b), detti terminali di ingresso e detti terminali di uscita in almeno due configurazioni di elaborazione,

in cui:

in una prima configurazione di elaborazione, detta circuiteria di calcolo (20) è configurata per calcolare detti dati di uscita (X0, X1) secondo un primo insieme di funzioni, e

in almeno una seconda configurazione di elaborazione, detta circuiteria di calcolo (20) è configurata per

calcolare detti dati di uscita (X_0, X_1) secondo un rispettivo secondo insieme di funzioni, detto rispettivo secondo insieme di funzioni essendo differente da detto primo insieme di funzioni.

2. Circuito (160) secondo la rivendicazione 1, comprendente:

rispettivi controllori di accesso diretto alla memoria in lettura configurabili ($200_0, 200_1$) accoppiati a un primo sottoinsieme di detto insieme di terminali di ingresso per ricevere (162, 163) un rispettivo primo sottoinsieme di detti segnali digitali di ingresso che trasportano un primo sottoinsieme di dati di ingresso (P, Q), in cui detti controllori di accesso diretto alla memoria in lettura configurabili ($200_0, 200_1$) sono configurati per controllare il prelevamento di detto primo sottoinsieme di dati di ingresso (P, Q) da una memoria (M_0, \dots, M_{Q-1}), e

rispettivi controllori di accesso diretto alla memoria in scrittura configurabili ($204_0, 204_1$) accoppiati a detto insieme di terminali di uscita per fornire (162, 163) detti segnali digitali di uscita che trasportano dati di uscita (X_0, X_1), in cui detti controllori di accesso diretto alla memoria in scrittura configurabili ($204_0, 204_1$) sono configurati per controllare la memorizzazione di detti dati di uscita (X_0, X_1) in detta memoria (M_0, \dots, M_{Q-1}).

3. Circuito (160) secondo la rivendicazione 2, comprendente rispettivi registri di buffer di ingresso ($202_0, 202_1$) accoppiati a detti controllori di accesso diretto alla memoria in lettura configurabili ($200_0, 200_1$) e rispettivi registri di buffer di uscita ($206_0, 206_1$) accoppiati a detti controllori di accesso diretto alla

memoria in scrittura configurabili (204₀, 204₁).

4. Circuito (160) secondo una qualsiasi delle rivendicazioni precedenti, comprendente:

un circuito generatore di indirizzi di ROM (208) configurato per controllare il prelevamento di un secondo sottoinsieme di dati di ingresso (W0) da almeno una memoria di sola lettura (164, 165) mediante un secondo sottoinsieme di detti segnali digitali di ingresso, e/o

un circuito generatore di indirizzi di memoria (210) configurato per controllare il prelevamento di un terzo sottoinsieme di dati di ingresso (W1, W2, W3, W4) da almeno una memoria configurabile locale (166, 167) mediante un terzo sottoinsieme di detti segnali digitali di ingresso.

5. Circuito (160) secondo una qualsiasi delle rivendicazioni precedenti, in cui detto insieme di circuiti moltiplicatori comprende un primo circuito moltiplicatore (30a), un secondo circuito moltiplicatore (30b), un terzo circuito moltiplicatore (30c) e un quarto circuito moltiplicatore (30d), detto insieme di circuiti sommatori-sottrattori (32a, 32b) comprende un primo circuito sommatore-sottrattore (32a) e un secondo circuito sommatore-sottrattore (32b), e detto insieme di circuiti accumulatori comprende un primo circuito accumulatore (34a) e un secondo circuito accumulatore (34b), e in cui:

detto primo circuito moltiplicatore (30a) riceve come primo operando un primo segnale di ingresso (W1) di detto insieme di segnali digitali di ingresso, e riceve come secondo operando un secondo segnale di ingresso (P) di detto insieme di segnali digitali di ingresso,

detto secondo circuito moltiplicatore (30b) riceve

come primo operando un terzo segnale di ingresso (Q) di detto insieme di segnali digitali di ingresso, e riceve come secondo operando un segnale selezionabile tra un quarto segnale di ingresso (W2) e un quinto segnale di ingresso (W4) di detto insieme di segnali digitali di ingresso,

detto terzo circuito moltiplicatore (30c) riceve come primo operando un segnale selezionabile tra un segnale di uscita da detto primo circuito moltiplicatore (30a) e detto secondo segnale di ingresso (P), e riceve come secondo operando un segnale selezionabile tra un sesto segnale di ingresso (W3), un settimo segnale di ingresso (W0) e detto quinto segnale di ingresso (W4),

detto quarto circuito moltiplicatore (30d) riceve come primo operando un segnale selezionabile tra un segnale di uscita da detto secondo moltiplicatore (30b) e detto terzo segnale di ingresso (Q), e riceve come secondo operando un segnale selezionabile tra detto quinto segnale di ingresso (W4) e detto settimo segnale di ingresso (W0),

detto primo circuito sommatore-sottrattore (32a) riceve come primo operando un segnale selezionabile tra detto segnale di uscita da detto primo circuito moltiplicatore (30a), detto secondo segnale di ingresso (P) e detto segnale di uscita da detto terzo circuito moltiplicatore (30c), e riceve come secondo operando un segnale selezionabile tra detto terzo segnale di ingresso (Q), detto segnale di uscita da detto secondo circuito moltiplicatore (30b) e un segnale nullo,

detto secondo circuito sommatore-sottrattore (32b) riceve come primo operando un segnale selezionabile tra detto segnale di uscita da detto terzo circuito moltiplicatore (30c) e detto segnale di uscita da detto

quarto circuito moltiplicatore (30d), e riceve come secondo operando un segnale selezionabile tra detto segnale di uscita da detto quarto circuito moltiplicatore (30d), detto segnale di uscita da detto secondo circuito moltiplicatore (30b) e un segnale nullo,

detto primo circuito accumulatore (34a) riceve come ingresso un segnale di uscita da detto primo circuito sommatore-sottrattore (32a),

detto secondo circuito accumulatore (34b) riceve come ingresso un segnale di uscita da detto secondo circuito sommatore-sottrattore (32b), e

detto primo circuito accumulatore (34a) è selettivamente attivabile (EN) per fornire un primo segnale di uscita (X0) e detto secondo circuito accumulatore (34b) è selettivamente attivabile (EN) per fornire un secondo segnale di uscita (X1).

6. Circuito (160) secondo la rivendicazione 5, in cui detta circuiteria di calcolo (20) comprende un insieme di circuiti configurati per calcolare funzioni non lineari (40a, 40b).

7. Circuito (160) secondo la rivendicazione 6, in cui detto insieme di circuiti configurati per calcolare funzioni non lineari (40a, 40b) comprende un primo circuito configurato per calcolare funzioni non lineari (40a) e un secondo circuito configurato per calcolare funzioni non lineari (40b), e in cui:

detto primo circuito configurato per calcolare funzioni non lineari (40a) riceve come ingresso un segnale di uscita da detto primo circuito accumulatore (34a),

detto secondo circuito configurato per calcolare

funzioni non lineari (40b) riceve come ingresso un segnale di uscita da detto secondo circuito accumulatore (34b),

detto primo segnale di uscita (X0) è selezionabile (36k) tra il segnale di uscita dal primo circuito accumulatore (34a) e il segnale di uscita dal primo circuito configurato per calcolare funzioni non lineari (40a), e

detto secondo segnale di uscita (X1) è selezionabile (36m) tra il segnale di uscita dal secondo circuito accumulatore (34b) e il segnale di uscita dal secondo circuito configurato per calcolare funzioni non lineari (40b).

8. Dispositivo (16) comprendente un insieme di circuiti (160) secondo una qualsiasi delle rivendicazioni precedenti, un insieme di banchi di memoria di dati (M_0, \dots, M_{Q-1}) e un'unità di controllo (161), in cui i circuiti (160) sono configurabili (161, 168) per leggere dati da e scrivere dati in detti banchi di memoria di dati (M_0, \dots, M_{Q-1}) mediante una rete di interconnessione (162, 163) in funzione di dati di configurazione memorizzati in detta unità di controllo (161).

9. Dispositivo (16) secondo la rivendicazione 8, in cui detti banchi di memoria di dati (M_0, \dots, M_{Q-1}) comprendono registri di buffer, preferibilmente registri di buffer doppi.

10. Sistema (1) comprendente un dispositivo (16) secondo la rivendicazione 8 o la rivendicazione 9 e un'unità di elaborazione (10) accoppiata al dispositivo (16) mediante un'interconnessione di sistema (18), in cui i

circuiti nell'insieme di circuiti (160) del dispositivo (16) sono configurabili in almeno due configurazioni di elaborazione in funzione di un segnale di controllo ricevuto dall'unità di elaborazione (10).

11. Procedimento di funzionamento di un circuito (160) secondo una qualsiasi delle rivendicazioni da 1 a 7, un dispositivo (16) secondo la rivendicazione 8 o la rivendicazione 9, o un sistema (1) secondo la rivendicazione 10, il procedimento comprendendo partizionare il tempo di funzionamento di detta circuiteria di calcolo (20) in almeno un primo e un secondo intervallo di funzionamento, in cui detta circuiteria di calcolo (20) è fatta funzionare rispettivamente in detta prima configurazione di elaborazione e in detta almeno una seconda configurazione di elaborazione.

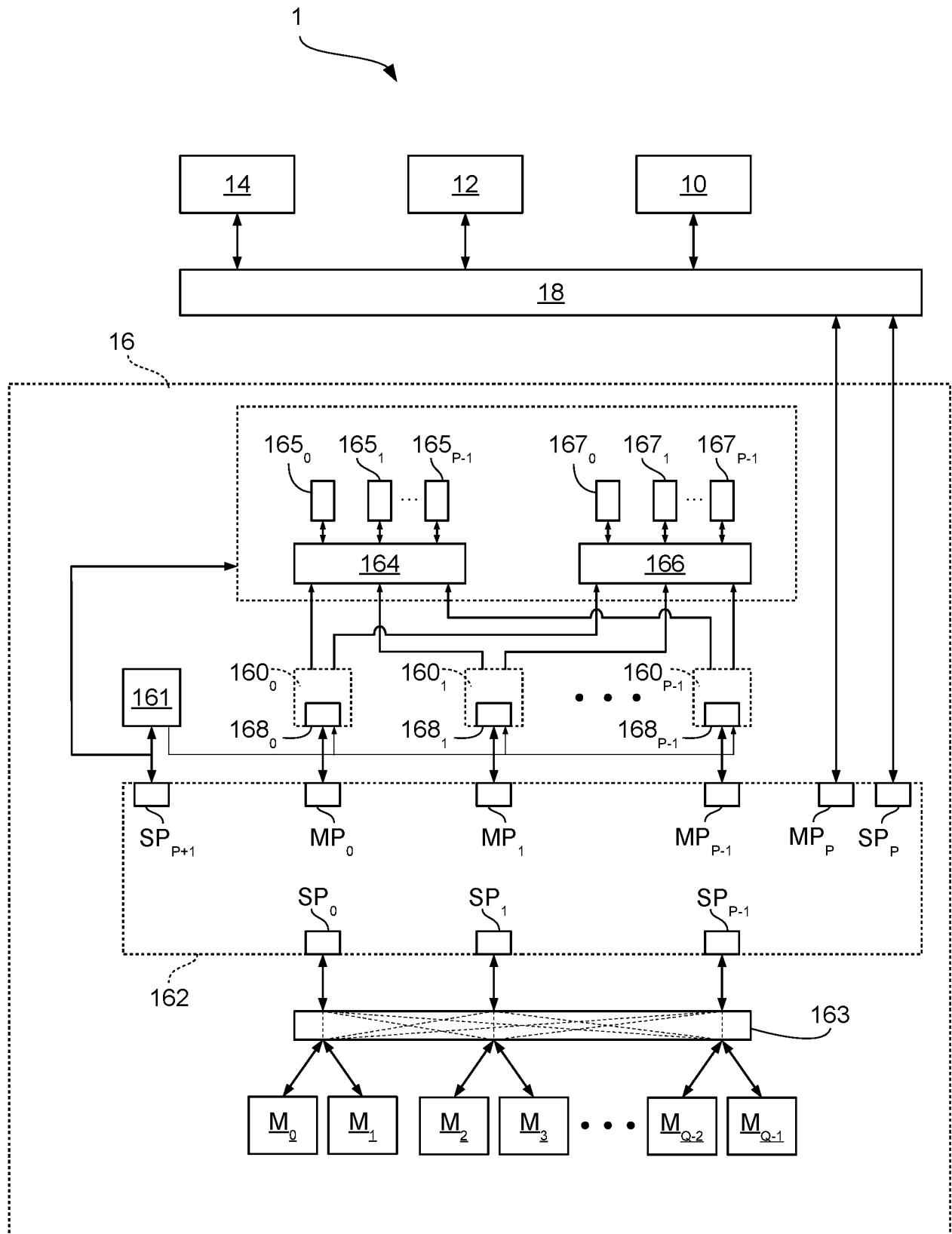
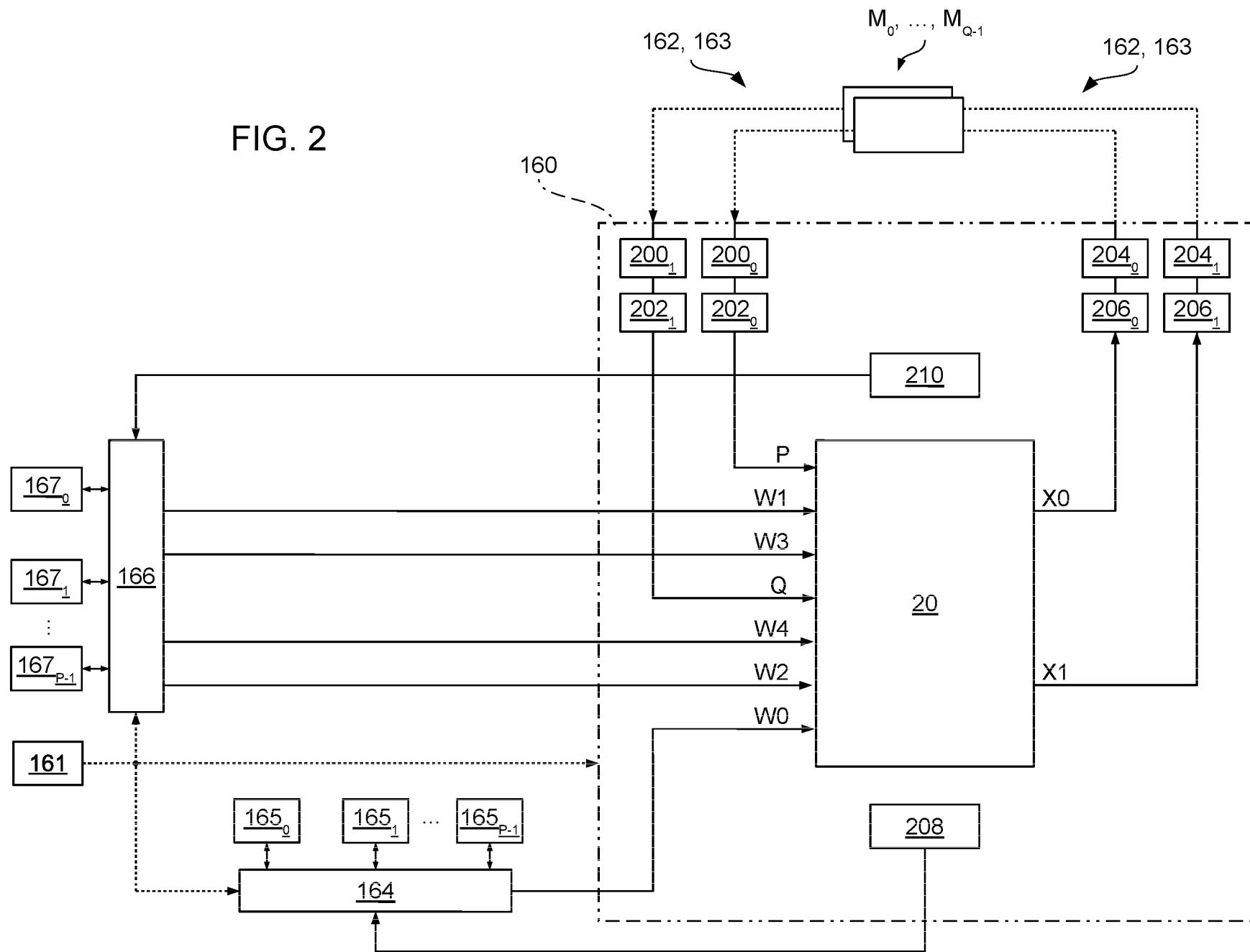


FIG. 1

FIG. 2



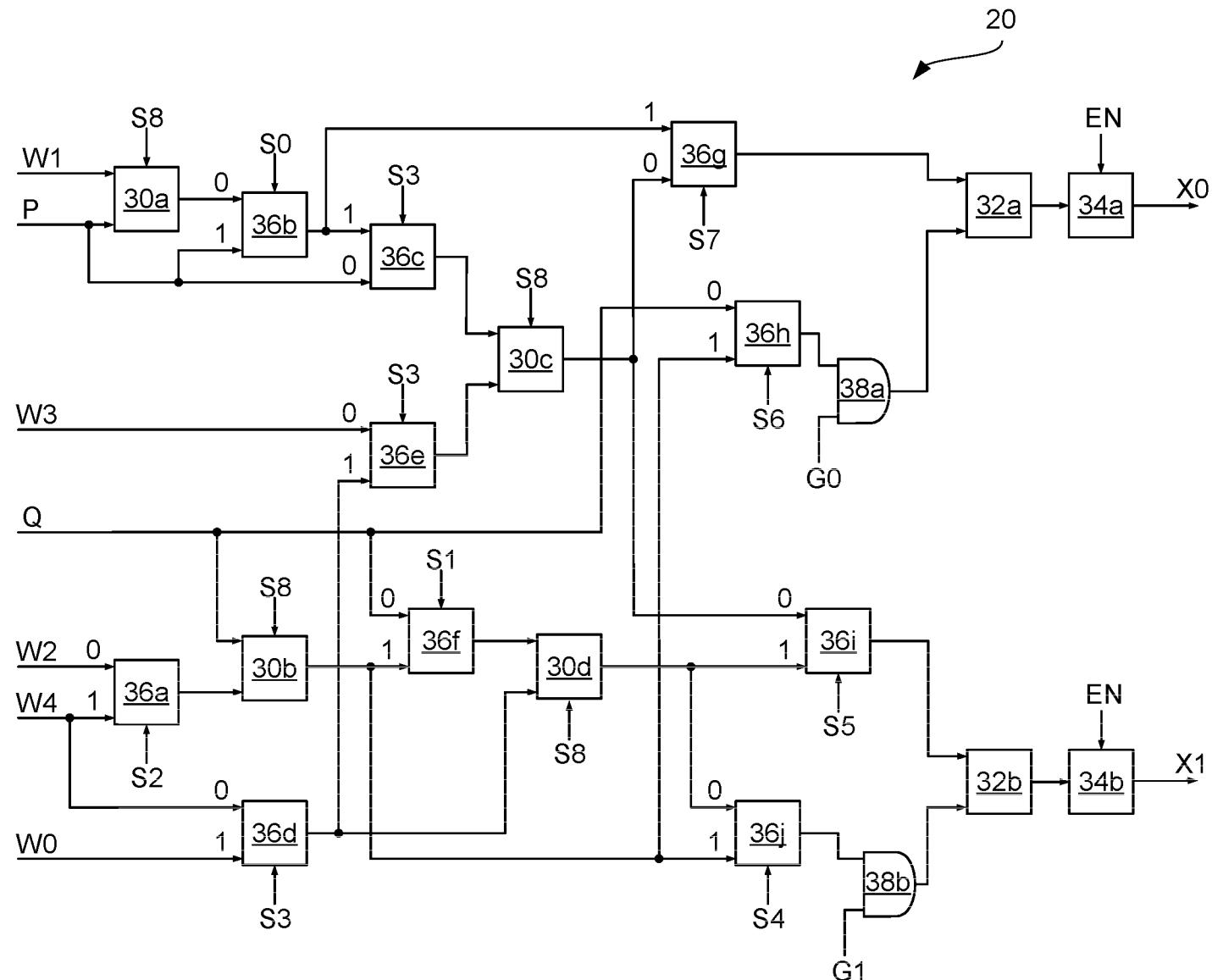


FIG. 3

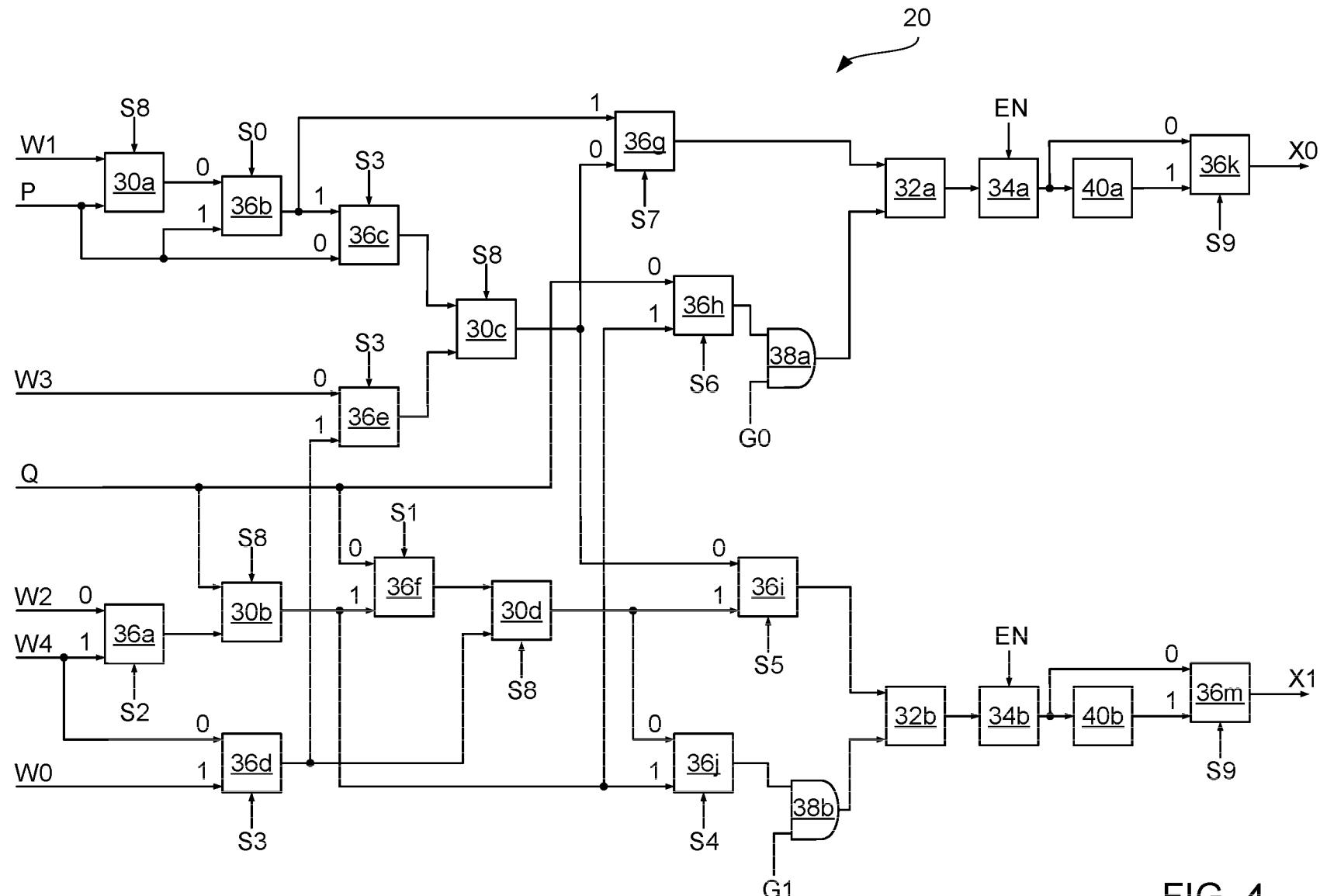


FIG. 4

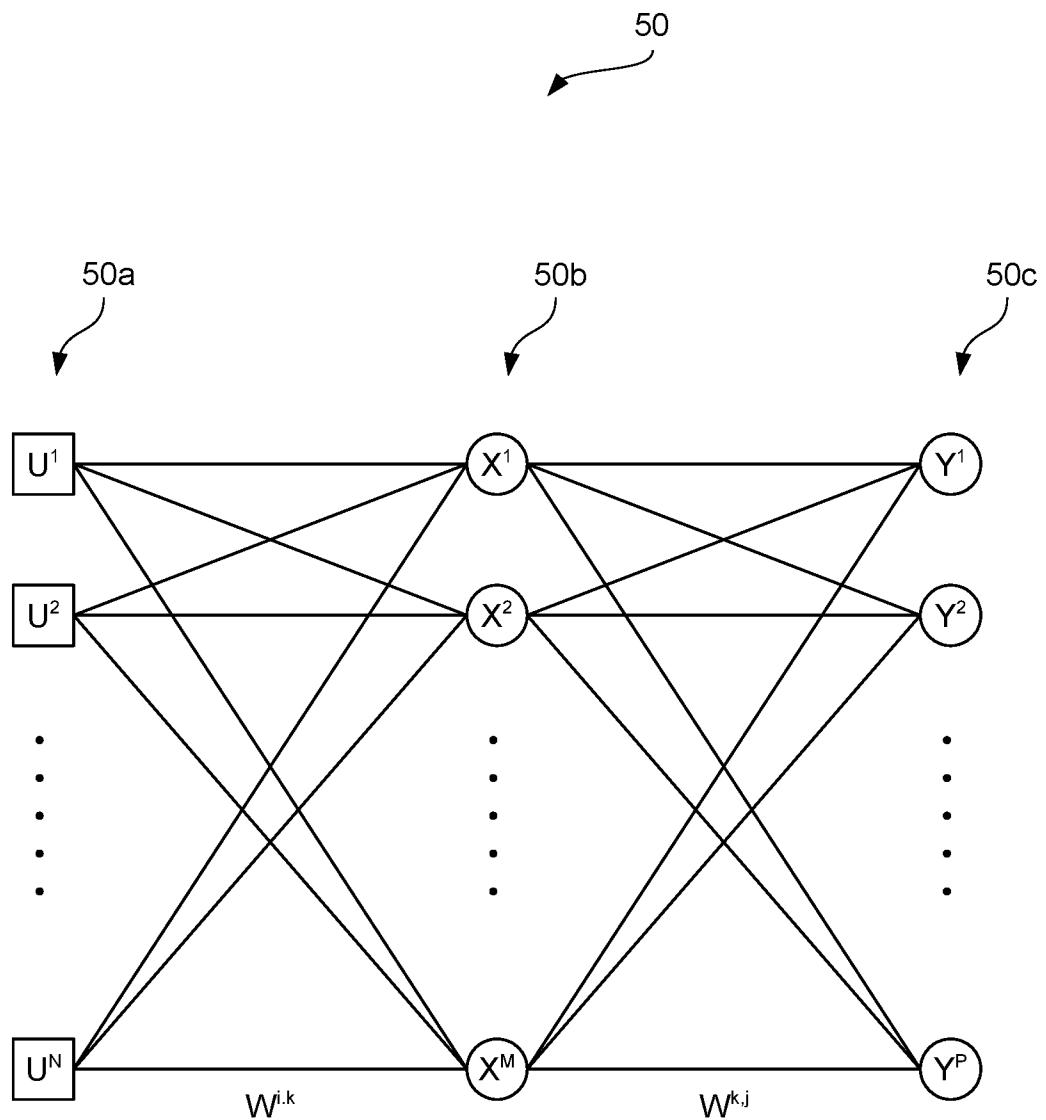


FIG. 5