

ITALIAN PATENT OFFICE

Document No.

102012902055597A1

Publication Date

20131130

Applicant

STMICROELECTRONICS S.R.L.

Title

PROCEDIMENTO PER GESTIRE TRANSAZIONI DI ACCESSO E RELATIVO
SISTEMA

DESCRIZIONE dell'invenzione industriale dal titolo:

"Procedimento per gestire transazioni di accesso e relativo sistema"

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

Inventori designati: Daniele MANGANO, Salvatore PISASALE e Mirko DONDINI

Depositata il: 30 maggio 2012

TESTO DELLA DESCRIZIONE

Campo tecnico

La presente descrizione si riferisce alle tecniche di gestione delle transazioni in sistemi per il trattamento dei segnali.

Varie forme di attuazione possono riferirsi a sistemi del tipo correntemente denominato System-on-Chip (SoC).

Sfondo tecnologico

Nella realizzazione di sistemi per il trattamento dei segnali, quali sistemi System-on-Chip o SoC, è pratica corrente integrare nel sistema nuclei (core) elaborativi quali, ad esempio, core ARM.

Tali nuclei elaborativi (ad esempio del tipo cortex-A9/15) possono prevedere l'impiego di protocolli, quali il protocollo AMBA (ad esempio AXI3/4), suscettibili di indurre specifici vincoli riguardo all'ordine in cui si possono svolgere determinate funzioni quali, ad esempio, le transazioni di accesso a memoria.

I sistemi basati su sistemi operativi quali ad es. Android possono integrare in varie soluzioni unità di gestione della memoria (Memory Management Unit o MMU) o funzioni similari che non permettono di controllare la allocazione dei buffer nella memoria.

Il ricorso a tecniche hardware avanzate per bilanciare i carichi di memoria fra più memorie (ad esempio del tipo Dual Data Rate o DDR) tende a rendere più complessi questi scenari, tenuta anche in conto la tendenza a fare sì che la complessità a livello hardware risulti mascherata rispetto alle applicazioni.

Gli inventori hanno notato che questo ha portato, ad esempio, a sviluppare specifici supporti hardware suscettibili, ad esempio, di assicurare direttamente il soddisfacimento dei vincoli di ordine (così come imposti, ad es. da protocolli quali AMBA); questo, ad esempio, facendo sì che l'hardware operi tenendo in conto gli indirizzi fisici, con la possibilità di assicurare anche la consistenza a livello di memoria richiesta da nuclei elaborativi quali ad esempio i nuclei ARM.

Lo schema a blocchi della figura 1 illustra a titolo di esempio un contesto applicativo del tipo qui considerato. Nello schema della figura 1, si vede un modulo 10 (si può trattare, ad esempio, di un blocco di logica del tipo correntemente indicato come nucleo o core Intellectual Property o IP) provvisto di driver 11 che gli permettono di interfacciarsi con un insieme di applicazioni APPS.

Nel funzionamento, il nucleo o core 10 (per fissare le idee - naturalmente senza alcun intento limitativo - si può pensare ad esempio ad una CPU) si trova ad accedere ad uno o più moduli "obiettivo" (moduli target) rappresentati, ad esempio, da moduli di memoria MEM1, MEM2, ecc. Si può trattare, ad esempio, di memorie del tipo di Dual Data Rate o DDR.

In varie soluzioni, l'accesso del modulo 10 ai moduli MEM1, MEM2, ecc. si può realizzare sotto forma di transazioni di accesso attuate con un meccanismo di

assegnazione di identificatori di transazione tramite una rete N. Si può trattare, ad esempio, di una cosiddetta ICN, termine spesso utilizzato come abbreviazione di "interconnect", ovvero del termine comunemente utilizzato per identificare i sistemi di comunicazione on-chip. Una forma avanzata di sistemi di comunicazione on-chip attualmente disponibile viene normalmente identificata come sistema Network-on-Chip o NoC.

A tal fine, in varie soluzioni, il modulo 10 può prevedere una funzione di assegnazione degli identificatori (ID assign) 12, operante ad esempio secondo un protocollo AMBA, e suscettibile di interfacciarsi, in varie soluzioni, con un'unità di gestione della memoria (Memory Management Unit o MMU). In varie soluzioni l'unità MMU può cooperare con la rete N tramite un modulo o unità di ri-mappatura degli indirizzi (Address Remapping o AR). In varie soluzioni, le unità MMU e AR possono svolgere, rispettivamente, funzioni classificabili come una funzione di astrazione della memoria (in dipendenza di una allocazione di buffer pilotata dalle applicazioni APPS) ed una funzione di bilanciamento del carico (verso le memorie DDR MEM1, MEM2, ...), dialogando fra loro per quanto riguarda la funzione di assegnazione del campo degli indirizzi fisici (Assign Physical Addresses Range).

L'unità MMU interagisce a livello di indirizzi virtuali con la funzione di assegnazione degli identificatori 12 del modulo 10 ed a livello di indirizzi fisici (Physical Address) con l'unità AR; quest'ultima interagisce con la rete N a livello di indirizzi ri-mappati (Remapped Addresses).

Secondo criteri generali applicabili quale che sia la natura dei moduli target (dunque non solo ad unità di

memoria quali le unità di memoria MEM1, MEM2,..... qui considerate a titolo di esempio) il funzionamento di un sistema - ad es. SoC - così come schematicamente rappresentato nella figura 1 può essere assoggettato a regole di ordinamento e di consistenza delle transazioni.

Ad esempio, in varie soluzioni, per transazioni aventi lo stesso identificatore si vuole poter assicurare un ordine nelle transazioni e fare in modo che le risposte possano essere fornite con lo stesso ordine delle richieste. Transazioni con identificatori diversi possono non essere assoggettate a particolari vincoli in termini di ordine (ad esempio due memorie MEM1 e MEM2 possono essere oggetto di accesso in parallelo senza violazioni del protocollo, cosa che in generale può non applicarsi a transazioni che hanno lo stesso identificatore). In varie soluzioni può essere previsto che una funzione di controllo delle memorie possa garantire un criterio di consistenza; questo, ad esempio, assicurando che accessi con lo stesso identificatore non siano rimescolati (reshuffled), così come per altri versi potrebbe essere vantaggioso fare per ottimizzare la efficienza delle memorie.

Gli inventori hanno osservato che un modo per conseguire il rispetto di regole di ordinamento/consistenza di questa natura può passare attraverso una funzione di filtraggio a livello hardware. Una soluzione di questo genere (nota con il nome commerciale di CDAS) è utilizzata in talune tecnologie di interconnect ARM (ad esempio quelle note come CoreLink NIC-301, CCI-400). Queste soluzioni si basano sull'impiego di una funzione hardware tale da evitare l'insorgere di situazioni di disordine incompatibili con il protocollo AXI. Ad esempio, una regola suscettibile di essere implementata è quella per cui, in

presenza di una transazione con un identificatore ID già assegnato ad un'altra transazione pendente (outstanding), la nuova transazione è bloccata aspettando che la transazione pendente "ritorni" (ossia sia compiuta) così da evitare di innescare situazioni di disordine.

Gli inventori hanno osservato che soluzioni di questo tipo possono peggiorare le prestazioni legate al tempo di attesa del ritorno della transazione pendente. Questa situazione può risultare particolarmente penalizzante nel caso di sistemi a più memorie (ad es. multi-DDR) con un meccanismo di ri-mappatura degli indirizzi per bilanciare il carico sulla memoria.

Gli inventori hanno altresì osservato che queste soluzioni non sono in generale applicabili ai sistemi SoC di tipo più recente, con integrati al loro interno nuclei elaborativi ARM. Questo in particolare in considerazione del fatto che le tecniche adottate per la virtualizzazione della memoria (MMU) e soprattutto per la ri-mappatura degli indirizzi con l'obiettivo di bilanciare il carico su più memorie fisiche, possono incidere negativamente, anche in misura rilevante, sulle prestazioni.

La figura 2 rappresenta, con immediato riferimento allo schema della figura 1 (parti od elementi identici o equivalenti a quelli già descritti con riferimento alla figura 1 non sono pertanto qui nuovamente descritti), uno schema di massima del tipo descritto in EP 2 444 903 A1.

La soluzione della figura 2 può prevedere la presenza di un'unità di riordino delle transazioni 20 tale da far sì che la catena di comunicazione fra il modulo 10 (ad es. una CPU) ed i moduli target (ad esempio le memorie MEM1 e MEM2) sia suddivisa in due sezioni:

- una sezione "ordinata", compresa fra il modulo 10 e

l'unità di riordino delle transazioni 20, e

- una sezione "disordinata", compresa fra il modulo di riordino delle transazioni 20 e i moduli target MEM1, MEM2.

In tale soluzione, è possibile assegnare identificatori diversi a transazioni che originariamente (prima dell'unità 20) hanno lo stesso identificatore ID.

Gli inventori hanno osservato che:

- con questa soluzione, la funzione di controllo dei moduli target MEM1, MEM2, ecc. (a livello di rete N) non è in grado di assicurare la consistenza nella gestione delle transazioni di accesso ai moduli target;

- nel caso di sistemi complessi, in grado di supportare un numero molto elevato di transazioni simultaneamente pendenti (outstanding) e/o un numero corrispondente elevato di moduli, in almeno alcune soluzioni possono insorgere problemi di saturazione degli identificatori ID.

Scopo e sintesi

Varie forme di attuazione si prefiggono lo scopo di superare i suddetti inconvenienti.

Secondo varie forme di attuazione, tale scopo è raggiunto grazie ad un procedimento avente le caratteristiche richiamate in modo specifico nelle rivendicazioni che seguono.

Varie forme di attuazione riguardano anche un corrispondente sistema, ad esempio a livello SoC (System-on-Chip).

Le rivendicazioni formano parte integrante dell'insegnamento tecnico qui somministrato in relazione all'invenzione.

Varie forme di attuazione possono basarsi su uno schema di ri-ordine compatibile con l'approccio di

consistenza della memoria richiesto da protocolli quali il protocollo AXI.

Varie forme di attuazione possono prevedere una specifica sequenza di passi che tengono in conto tanto il modulo target cui si accede quanto le condizioni di sistema.

Varie forme di attuazione possono assicurare una consistenza e prestazioni elevate anche nel caso di sistemi SoC che integrino nuclei ARM a livello di system embedded.

Varie forme di attuazione consentono di utilizzare tecniche di gestione della memoria avanzate in sistemi a memorie multiple (ad esempio multi-DDR) senza che ciò incida negativamente sulle prestazioni e/o in vincoli stringenti in termini di sistema.

Varie forme di attuazione possono mettere in atto un meccanismo di compressione dei bit degli identificatori di transazione così da evitare fenomeni di saturazione delle mappe degli identificatori.

Breve descrizione delle figure

L'invenzione sarà ora descritta, a puro titolo di esempio non limitativo, con riferimento alle figure annesse, in cui:

- le figure 1 e 2 sono già state descritte in precedenza,
- la figura 3 è uno schema a blocchi di massima di forme di attuazione,
- la figura 4 è un diagramma di flusso rappresentativo di possibili criteri di funzionamento di forme di attuazione,
- la figura 5 è uno schema a blocchi di dettaglio di forme di attuazione,
- la figura 6 illustra a livello simbolico un effetto

di compressione suscettibile di essere messo in varie forme di attuazione, e

- la figura 7 è un ulteriore schema a blocchi di dettaglio di forme di attuazione.

Descrizione particolareggiata

Nella seguente descrizione sono illustrati vari dettagli specifici finalizzati ad un'approfondita comprensione di varie forme di attuazione. Le forme di attuazione possono essere realizzate senza uno o più dei dettagli specifici, o con altri metodi componenti materiali, etc. In altri casi, strutture, materiali o operazioni noti non sono mostrati o descritti in dettaglio per evitare di rendere oscuri i vari aspetti delle forme di attuazione.

Il riferimento ad "una forma di attuazione" nell'ambito di questa descrizione sta ad 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", eventualmente presenti in diversi luoghi di questa descrizione non sono necessariamente riferite alla stessa forma di attuazione. Inoltre, particolari conformazioni, strutture o caratteristiche possono essere combinate in ogni modo adeguato in una o più forme di attuazione.

I riferimenti qui utilizzati sono soltanto per comodità e non definiscono dunque l'ambito di tutela o la portata delle forme di attuazione.

In termini di architettura complessiva, varie forme di attuazione così come esemplificate nella figura 3 possono rifarsi allo scenario ed allo schema generale già presentato con riferimento alle figure 1 e 2. La

descrizione fornita in precedenza riguardo a tali figure deve quindi ritenersi applicabile anche alle forme di attuazione descritte con riferimento alle figure 3 a 7 salvo per quanto diversamente descritto in relazione a queste ultime figure.

Nello schema della figura 3 non è riprodotta la parte della figura 2 relativa al collegamento con i moduli target MEM1, MEM2 tramite la rete N. Questo fatto evidenzia ulteriormente il carattere del tutto esemplificativo del riferimento in precedenza fatto a moduli di memoria e ad un collegamento attuato tramite una rete ICN.

Il contesto cui si riferiscono le forme di attuazione è dunque, in generale, quello di un sistema di elaborazione comprendente almeno un modulo 10 (qui esemplificato da un nodo o core IP quale ad esempio una CPU) che accede attraverso un meccanismo basato su transazioni di accesso ad una pluralità di moduli obiettivo o target (ad esempio le memorie MEM1, MEM2).

In varie forme di attuazione, il meccanismo di accesso può basarsi sul fatto che alle suddette transazioni vengono assegnati rispettivi identificatori di transazione.

Varie forme di attuazione possono prevedere che, ad esempio a livello dell'unità di riordino delle transazioni indicata con 20, gli identificatori destinati ad essere assegnati alle transazioni di accesso ad un dato modulo o target siano sottoposti (quali identificatori d'ingresso ID) ad un controllo o check di consistenza.

In varie forme di attuazione, tale controllo o check può basarsi sul criterio di fondo di implementare uno schema di ri-ordino delle transazioni compatibile con l'approccio di consistenza che può essere richiesto (ad esempio da un protocollo quale il protocollo AXI) a livello

di moduli target (ad esempio le memorie MEM1 e MEM2).

In varie forme di attuazione, suddetta operazione di controllo o check può operare sulla base di un meccanismo suscettibile di essere espresso secondo una relazione funzionale f del tipo:

$$CID = f(ID, TRG)$$

dove:

- ID è l'identificatore "di ingresso" assegnato ad una data transazione,
- TRG identifica il modulo target cui si intende accedere con la transazione in questione,
- CID è un identificatore consistente "di uscita" assegnato alla coppia ID/TRG secondo i criteri meglio descritti nel seguito.

Si apprezzerà che l'approccio qui descritto può essere applicato a un numero qualsiasi di moduli target (TRG) in modo indipendente dalla natura del singolo modulo.

In varie forme di attuazione, l'identificatore CID può essere un identificatore consistente in quanto soddisfa un criterio di consistenza tale per cui le transazioni di accesso non vengono rimescolate ("reshuffled") e dunque conservano il loro ordine, "ritornando" nello stesso ordine con cui sono state emesse.

In coerenza con criteri generali adottati nel definire la l'architettura dei sistemi di elaborazione, in varie forme di attuazione un identificatore consistente può essere un identificatore che conserva l'ordine di dette transazioni. Protocolli quali AMBA AXI possono infatti prevedere (ad esempio tramite il cosiddetto "ordering model" del protocollo AMBA AXI) che le transazioni con lo stesso identificativo non vengano rimescolate dai moduli target. Questa regola permette di garantire la consistenza

dei dati, in altre parole facendo in modo che il dato scritto in corrispondenza di un indirizzo di memoria sia quello corrispondente all'ultima scrittura effettuata sullo stesso indirizzo.

Varie forme di attuazione permettono quindi di assicurare che in tutti i casi in cui è richiesta consistenza (ovvero esistono transazioni con lo stesso ID destinate allo stesso modulo target) le transazioni continuino ad obbedire alla regola di protocollo atta a garantire la consistenza, permettendo quindi di far sì che queste siano ricevute dal target con lo stesso ID, anche se di per sé diverso da quello di partenza.

Così come qui utilizzato, il termine "consistente" è dunque legato alla definizione di consistenza dei dati in memoria ed indica che le transazioni hanno ID tali da permettere di assicurare la consistenza in memoria. In varie forme di attuazione, un modulo target dato TRG può pertanto costituire una coppia identificatore(ID)/modulo target dato (TRG) cui si assegna un identificatore di uscita consistente (CID) che permette quindi di conservare l'ordine delle transazioni.

Ad esempio, in varie forme di attuazione, l'identificatore CID può essere in grado di soddisfare un vincolo di ordine così come imposto, ad esempio, da un protocollo come AMBA e/o essere in grado di assicurare la consistenza di memoria così come richiesta da nuclei elaborativi come i nuclei ARM.

Varie forme di attuazione possono basarsi sul criterio di base in base al quale se lo stesso identificatore "di ingresso" ID è già stato emesso, in particolare per lo stesso modulo target TRG (con la transazione o le transazioni associate pendenti), allora come identificatore

consistente "di uscita" CID si (ri)utilizza lo stesso identificatore "di ingresso" ID.

Varie forme di attuazione possono così assicurare che transazioni aventi originariamente lo stesso identificatore ID sono indirizzate allo stesso modulo target conservando lo stesso identificatore ID anche dopo il controllo di consistenza, così da poter soddisfare regole di consistenza alla base dell'accesso ai moduli target (così come dettate, ad esempio, da un protocollo come AXI).

In varie forme di attuazione, il suddetto procedimento di (ri)ordine delle transazioni può realizzarsi a livello dell'unità 20 secondo l'impostazione esemplificata dal diagramma di flusso della figura 4.

In tale diagramma si assume che il passo 100 sia un passo iniziale corrispondente alla generazione di una nuova transazione destinata ad un modulo target dato (TRG, con i formalismo adottato nella relazione in precedenza richiamata) cui inizialmente è stato assegnato quale identificatore di "ingresso" in un indicatore ID. In varie forme di attuazione, i criteri di assegnazione di un tale identificatore di ingresso possono corrispondere a criteri noti. In varie forme di attuazione, per l'assegnazione degli identificatori di ingresso ID si può far ricorso ai criteri meglio descritti nel seguito.

A partire dal passo iniziale 100, in un passo 102 l'unità 20 verifica se l'identificatore di ingresso ID è già stato emesso in precedenza.

Se la verifica del passo 102 dà esito positivo (Y=YES), in un passo 104 si verifica se l'identificatore ID in questione, verificato essere già stato emesso, è stato emesso per lo stesso modulo target (TRG) cui si riferisce la nuova transazione. Se anche la verifica del passo 104 dà

esito positivo (Y=YES), l'unità 20 procede, in un passo 106, a riutilizzare - come identificatore CID assegnato come identificarono "consistente" di uscita alla coppia ID/TRG - l'identificatore ID ricevuto in ingresso.

Se, al contrario, uno dei passi 102 e 104 dà esito negativo (N=NO), indizio del fatto che l'identificatore ID in ingresso non è stato emesso in precedenza (esito negativo del passo 102) oppure che l'identificatore ID in ingresso, pur essendo già stato emesso è stato emesso per un modulo target TRG diverso da quello cui si riferisce la transazione in ingresso (esito negativo del passo 104), in un passo 108 si procede (ad esempio secondo i criteri meglio descritti nel seguito) ad assegnare alla coppia ID/TRG un identificatore "consistente" CID diverso da ID.

La procedura di attribuzione di un identificatore "consistente" CID può quindi dirsi completata in un passo 110 cui, in varie forme di attuazione, possono seguire ulteriori operazioni così come meglio esemplificate nel seguito.

Varie forme di attuazione possono dunque prevedere che, data una coppia identificatore ID/modulo target TRG in ingresso, a tale coppia sia accoppiato un identificatore consistente CID identificato secondo i seguenti criteri:

- se l'identificatore in ingresso ID è già stato emesso per lo stesso modulo target TRG, l'identificatore ID in ingresso viene conservato come identificatore consistente CID in uscita,

- se l'identificatore in ingresso ID non è stato ancora emesso oppure è già stato emesso ma in relazione ad un modulo target TRG diverso, alla coppia identificatore ID/modulo target TRG in ingresso si assegna un (nuovo) identificatore d'uscita consistente CID, diverso

dall'identificatore in ingresso ID.

Così come esemplificato nella figura 5, in varie forme di attuazione, l'unità 20 può comprendere, disposti in cascata a partire andando da un lato di ingresso (rivolto verso il modulo 10) ad un lato d'uscita (rivolto verso la rete N) quattro moduli o blocchi:

- un modulo ripartitore o splitter 202, con la funzione di generare identificatori ID compresi in campi distinti per ciascuno dei moduli target,

- un modulo compressore 204 con la funzione di svolgere la verifica esemplificata con riferimento al diagramma di flusso della figura 4 e di attribuire alle copia ID/TRG gli identificatori CID "consistenti": la designazione di questo modulo quale modulo compressore tiene in conto il fatto che, adottando il criterio di varie forme di attuazione, il numero di identificatori totale è sempre pari o inferiore rispetto al numero delle transazioni pendenti,

- un modulo di riordino 206, con la funzione di riordinare gli identificatori consistenti CID accoppiando agli stessi etichette o tag TID secondo i criteri meglio descritti nel seguito, anche ai fini del riordino delle risposte, e

- un modulo di inseguimento o tracker 208 con la funzione è di gestire le transazioni di accesso sfruttando la gerarchia di riordino così legata agli indicatori CID.

In varie forme di attuazione, il modulo ripartitore o splitter 202 può operare generando gli identificatori ID aggiungendo ad identificatori portati al suo ingresso un offset che è funzione del modulo target TRG cui di volta in volta si desidera accedere.

In varie forme di attuazione, l'offset in questione

può essere definito secondo una relazione del tipo:

$$\text{offset} = \text{TRG} * k$$

dove k è una costante, che può essere scelta pari al numero (massimo) di identificatori che si desidera poter attribuire per ciascun modulo target nel sistema e TRG è un valore $(0, 1, \dots, N_b - 1)$ che identifica il modulo target preso in considerazione, dove N_b identifica il numero di moduli target presenti nel sistema.

Operando secondo questi criteri, il modulo 202 permette all'unità 20 di far sì che gli identificatori consistenti CID assegnati secondo i criteri in precedenza descritti con riferimento al diagramma di flusso della figura 4 possano identificare la coppia identificatore/modulo target portata in ingresso.

In varie forme di attuazione, il modulo splitter 202 è in grado di identificare a quale modulo target si riferisce ciascuna transazione trattata in funzione di indirizzi ricevuti, ad esempio, dai moduli MMU e AR, così come schematicamente rappresentato in 202a della figura 5. In varie forme di attuazione è possibile configurare staticamente o programmare in registri regioni di memoria relative ai vari moduli target.

In varie forme di attuazione, così come schematicamente rappresentato nello schema della figura 5 ciascuno dei moduli 202, 204, 206, 208 può essere realizzato in modo da realizzare:

- una determinata funzione "in avanti" - così come qui descritta - nel verso "in avanti" dal modulo 10 verso i moduli target MEM1, MEM2, ecc. e

- una funzione complementare "all'indietro" nel verso opposto (ossia nel verso delle "risposte" provenienti dai moduli target MEM1, MEM2, ecc. e dirette verso il modulo

10): le caratteristiche di una tale funzione complementare, quando qui non espressamente richiamate, risultano in modo diretto dalla descrizione della funzione "in avanti" qui fornita.

Ad esempio, in fase di risposta, il modulo 202 può togliere l'offset TRG*k aggiunto nella operazione "in avanti", così da ripristinare il valore originario dell'identificatore ID da restituire verso il modulo 10.

In varie forme di attuazione, il modulo compressore 204 può corrispondere alla soluzione già descritta in WO 2011/095963 A1 in funzione di una operazione di rimappatura degli identificatori.

In varie forme di attuazione, tale soluzione si può basare sull'impiego di una tabella (ad esempio una Look Up Table o LUT) le cui voci o entry sono costituite dalle coppie identificatore ID/modulo target TRG corrispondenti a transazioni pendenti (outstanding), cui può essere associato un valore di conteggio indicativo di quante transazioni pendenti corrispondono a ciascuna coppia.

Il numero massimo di coppie può essere assunto pari ad un valore definibile come MOT mentre il numero massimo di transazioni pendenti per coppia può essere identificato dal valore MOTC, tali valori potendo, in varie forme di attuazione, essere definiti dall'esterno del modulo così come schematicamente rappresentato in 204a (MOT) e 204b (MOTC).

A ciascuna voce o entry della tabella può essere associato, secondo i criteri descritti in precedenza, un corrispondente identificatore consistente CID e ciascuna voce può essere riempita con una coppia arbitraria di identificatore ID/modulo target TRG.

In varie forme di attuazione è possibile prevedere che

quando la tabella in questione è piena, il traffico (ossia la generazione/trattamento di nuove transazioni) sia interrotto, attendendo il ritorno di risposte per le transazioni pendenti.

Secondo i criteri già richiamati in precedenza, in particolare con riferimento al diagramma di flusso della figura 4, in varie forme di attuazione, se un identificatore ID ricevuto in ingresso dal modulo 204 si trova già nella suddetta tabella (per lo stesso modulo target, condizione riconoscibile, ad esempio, per effetto dell'offset aggiunto nel modulo splitter 202), è possibile ri-utilizzarlo come identificatore consistente CID incrementando di un'unità il valore del conteggio corrispondente.

In varie forme di attuazione, il riferimento ad un identificatore ID ricevuto in ingresso al modulo 204, può in realtà applicarsi al valore $(TRG * k + ID)$, ovvero al valore generato dal modulo 202 a partire delle due informazioni di partenza: ID e Target cui si accede.

In varie forme di attuazione, questa situazione può assumere il carattere di una vera e propria regola: se, all'arrivo di una nuova transazione, esistono già transazioni pendenti con lo stesso $(TRG * k + ID)$ (o equivalentemente coppia ID/TRG), il corrispondente CID viene riutilizzato al fine di assicurare la consistenza.

In varie forme di attuazione, il conteggio può dunque riferirsi al numero di transazioni pendenti per $(TRG * k + ID)$, e dunque coppia ID/TRG o, in definitiva, CID.

In varie forme di attuazione, si può procedere a generare un nuovo identificatore consistente CID riempiendo la corrispondente voce nella tabella con una coppia identificatore d'ingresso ID/modulo target TRG scelta in

modo che può essere di per sé arbitrario, inizializzando ad uno il relativo conteggio.

In varie forme di attuazione, una possibilità per la scelta del nuovo CID da assegnare può essere quella di popolare la prima riga vuota della tabella (ovvero la prima riga con conteggio pari a 0) e di utilizzare l'indice della riga come valore di CID.

In varie forme di attuazione, ogni volta che un identificatore consistente CID "ritorna" a partire dai moduli target, indizio del fatto che la relativa transazione di accesso è conclusa, ossia non più pendente, il conteggio ad esso associato può essere decrementato di un'unità. In varie forme di attuazione, se il contatore assume valore zero, questo può significare che non ci sono più transazioni pendenti per un certo CID, il che significa che il relativo identificatore consistente CID può essere utilizzato per un'altra coppia identificatore ID/modulo target TRG in ingresso.

Lo schema della figura 6 evidenzia l'effetto di "compressione" attuato dal modulo 204 (anche in funzione dell'azione del modulo splitter 202), tale da far sì che il valore del numero massimo MOT di coppie risulti inferiore rispetto al numero degli identificatori ID).

In particolare, lo schema della figura 6 fa vedere che partendo da identificatori compresi nel campo 0:K-1 e riferendosi, a titolo di esempio al caso in cui siano presenti due moduli target TRG 1 e TRG 2, la funzione del modulo splitter 202 porta a definire per il primo modulo target TRG=1 valori di identificatori ID compresi fra 0:K-1 e per il secondo target TRG=2 valori di identificatori ID compresi fra K e 2K-1. Il meccanismo di (ri)utilizzo degli identificatori ID già assegnati per lo stesso modulo

target TRG fa sì che gli identificatori consistenti CID possano assumere un numero di valori distinti compresi fra 0 e MOT-1.

Così come già detto in precedenza, in senso opposto (ossia nel verso della risposta) il modulo 204 potrà realizzare una funzione complementare, ossia restituire, per ciascun identificatore CID "di ritorno" l'identificatore ID corrispondente (compreso l'offset $K \cdot TRG$, destinato ad essere rimosso dal modulo 202).

In varie forme di attuazione, anche per il modulo 206 è possibile ricorrere ad una configurazione hardware proposta ad esempio in EP 2 444 903 A1 (già citato in precedenza, dove viene descritta una soluzione utilizzabile per un'azione generale di riordino di identificatori di transazione.

In varie forme di attuazione, il modulo 206 può operare aggiungendo a ciascuno degli identificatori consistenti CID ricevuti in ingresso una etichetta o tag di riordino TID prelevato ad esempio da un blocco 206a contenente etichette o tag in numero massimo pari a TMOT (numero corrispondente in pratica al numero massimo di transazioni che possano risultare pendenti nel sistema considerato).

In varie forme di attuazione, per ciascun identificatore consistente CID ricevuto al suo ingresso, il modulo 206 può emettere in uscita una coppia CID + TID comprendente l'identificatore consistente CID in associazione con una corrispondente etichetta TID.

Anche in questo caso, così come già detto in precedenza, in senso opposto (ossia nel senso della risposta) un modulo 206 può realizzare una funzione complementare, di rimozione delle etichette o tag TID dalla

coppia.

In varie forme di attuazione, le etichette o Tag TID in questione possono essere utilizzate per realizzare un'azione di riordino secondo uno schema compatibile con il funzionamento dei moduli target determinato dai protocolli in uso (ad esempio AXI). Si apprezzerà che il numero massimo TMOT di transazioni pendenti suscettibile di essere gestite è maggiore o al più uguale al valore MOT che identifica il numero massimo di coppie nella tabella del modulo compressore 204.

In varie forme di attuazione, la suddetta azione di riordino può essere svolta dal modulo 208, che ha disponibili in ingresso anche i valori MOT e MOTC così come schematicamente rappresentato in 208a 208b.

In varie forme di attuazione, il modulo 208 può operare secondo lo schema funzionale esemplificato nella figura 7.

Ad esempio, in varie forme di attuazione, il modulo 208 è in grado di raccogliere in corrispondenti code ad esempio di tipo FIFO (First In First Out):

- da una parte, le richieste presentate sotto forma di identificatori consistenti CID, e
- dall'altra parte le corrispondenti richieste di assegnazione di etichetta o tag TID.

Il tutto per poi restituire "all'indietro" tanto risposte in forma di etichette TID, quanto risposte in forma di identificatori CID.

In varie forme di attuazione, il modulo 208 può operare facendo sì che per un dato identificatore consistente CID, le etichette o tag TID siano raccolte in una coda FIFO. Questo modo di procedere è possibile in considerazione del fatto che esiste, proprio per come gli

stessi sono stati realizzati, un ordine per gli identificatori CID.

In varie forme di attuazione, ciò può corrispondere proprio alla regola di protocollo AMBA AXI atta a garantire la consistenza ("ordering model" di AMBA AXI). In varie forme di attuazione, l'ordine per CID può essere quindi garantito da protocollo e, grazie a questa ipotesi, risulta possibile utilizzare lo schema di figure 7 con l'obiettivo di conservare localmente i tag di riordino necessari in risposta al modulo 206 per realizzare la funzione di riordino vera e propria.

La profondità delle code FIFO suscettibili di essere implementate tramite il modulo 208 corrisponde al numero massimo di transazioni per ciascun identificatore CID (ossia il valore MOTC) ed il numero di code, ad esempio FIFO, può corrispondere al valore MOT.

Potendoci essere più transazioni con lo stesso identificatore consistente CID, il numero di transazioni pendenti (outstanding) supportate dal sistema può essere maggiore del valore MOT (consistendo in pratica nel valore garantito).

Varie forme di attuazione prevedono quindi una funzione utilizzabile per ri-mappare gli identificatori ID in una forma specifica tale da permettere un riordino consistente con le modalità di funzionamento di moduli target così come definiti, ad esempio a livello di protocollo (ad esempio AMBA). In varie forme di attuazione, le relative funzioni possono essere implementate a livello hardware, eventualmente utilizzando componenti hardware già noti e descritti in precedenti documenti.

Varie forme di attuazione permettono di assicurare una consistenza di funzionamento dei moduli target (ad esempio

nel caso di memorie quali ad esempio memorie DDR), ad esempio per sistemi del tipo System-on-Chip (SoC) in cui sono integrati (embedded) nuclei elaborativi ARM, garantendo un elevato livello di prestazioni.

Varie forme di attuazione permettono di utilizzare tecniche avanzate di gestione delle memorie, ad esempio in sistemi di memoria multi-DDR, senza influenzare negativamente le prestazioni e/o imporre vincoli rilevanti sul sistema.

In varie forme di attuazione, la compressione dei bit degli identificatori ID può permettere di evitare la saturazione della relativa mappa degli identificatori.

Varie forme di attuazione permettono di conseguire uno o più dei seguenti vantaggi:

- eliminazione di vincoli a livello software,
- livello di astrazione più elevato, con maggiore flessibilità a livello di sistema operativo e di impiego delle unità di pilotaggio,
- applicabilità a qualunque sistema che incorpori nuclei ARM,
- possibilità di applicare un approccio corrispondente in tecnologie di interconnect di tipo diverso, ad esempio ARM, Arteris, Sonics, ecc.

Naturalmente, fermo restando il principio dell'invenzione, i particolari di realizzazione le forme di attuazione potranno variare, anche in modo significativo, rispetto a quanto qui illustrato a puro titolo di esempio non limitativo, senza per questo uscire dall'ambito di protezione; tale ambito di protezione è definito dalle rivendicazioni.

RIVENDICAZIONI

1. Procedimento per gestire le transazioni di accesso di almeno un modulo (10) ad un modulo obiettivo o target dato (TRG) in una pluralità di moduli obiettivo o target (MEM1, MEM2) in un sistema per l'elaborazione di segnali, il procedimento comprendendo assegnare a dette transazioni rispettivi identificatori di transazione (ID, CID) e sottoporre detti identificatori di transazione ad una verifica (102, 104) per verificare se un identificatore (ID) in ingresso alla verifica (102, 104) è già stato emesso, detto identificatore (ID) in ingresso alla verifica (102, 104) e detto modulo target dato (TRG) costituendo una coppia identificatore(ID)/modulo target dato (TRG) cui si assegna un identificatore consistente in uscita (CID), in cui:

- i) se detto identificatore (ID) in ingresso alla verifica (102, 104) è già stato emesso per uno stesso modulo target dato (TRG), assegnare (106) a detta coppia identificatore(ID)/modulo target dato (TRG), quale identificatore consistente in uscita (CID), detto identificatore in ingresso (ID), e

- ii) se detto identificatore in ingresso (ID) alla verifica non è stato già emesso (102) o è già stato emesso (104) per un modulo target diverso da detto modulo target dato (TRG), assegnare (108) a detta coppia identificatore(ID)/modulo target dato (TRG), quale identificatore consistente in uscita (CID), un nuovo identificatore.

2. Procedimento secondo la rivendicazione 1, comprendente generare (202) detti identificatori in ingresso (ID) in campi disgiunti di identificatori, un

campo per ciascun modulo target (TRG) in detta pluralità di moduli target (MEM1, MEM2).

3. Procedimento secondo la rivendicazione 2, comprendente generare (202) gli identificatori in ingresso (ID) per un rispettivo modulo target (TRG) di detta pluralità di moduli target (MEM1, MEM2) come somma di un valore di identificatore dato ed un offset $K \cdot \text{TRG}$ dove K è un valore costante e TRG è un intero che identifica detto rispettivo modulo target in detta pluralità di moduli target (MEM1, MEM2).

4. Procedimento secondo una qualsiasi delle rivendicazioni 1 a 3, comprendente:

- mantenere una tabella avente come voci (entry) le coppie identificatore/modulo target dato (TRG) corrispondenti alle transazioni di accesso pendenti verso detta pluralità di moduli target (MEM1, MEM2), ciascuna coppia avendo associato un conteggio delle corrispondenti transazioni di accesso pendenti,

in cui:

- i) se un identificatore in ingresso (ID) a detta verifica (102, 104) è contenuto in una voce di detta tabella, assegnare detto identificatore in ingresso (ID) come identificatore consistente in uscita (CID) a detta voce della tabella ed incrementare di un'unità detto conteggio associato, e

- ii) se un identificatore in ingresso (ID) a detta verifica (102, 104) non è contenuto in una voce di detta tabella, riempire una voce vuota in detta tabella ed inizializzare un corrispondente conteggio associato per detta voce vuota che viene riempita.

5. Procedimento secondo la rivendicazione 4, comprendente verificare se detta tabella è piena e, se la

tabella è piena, inibire ulteriori transazioni di accesso verso detta pluralità di moduli target (MEM1, MEM2).

6. Procedimento secondo la rivendicazione 4 o la rivendicazione 5, comprendente il ridurre di un'unità detto conteggio associato ogniqualevolta una corrispondente transazione di accesso è stata completata.

7. Procedimento secondo una qualsiasi delle rivendicazioni 4 a 6, comprendente il verificare se detto conteggio è a zero e, se a zero, rendere il corrispondente identificatore consistente (CID) disponibile per l'assegnazione ad una nuova transazione.

8. Procedimento secondo una qualsiasi delle precedenti rivendicazioni, comprendente:

- accoppiare a ciascuna di dette transazioni di accesso ed all'identificatore consistente in uscita (CID) ad essa assegnato una etichetta o tag di riordino (TID), e
- propagare verso detti moduli target (MEM1, MEM2) cui detta transazione fa accesso tanto l'identificatore in uscita consistente (CID) ad essa assegnato quanto l'etichetta o tag (TID) ad essa accoppiata, per cui le transazioni di ritorno da detti moduli target (MEM1, MEM2) sono ri-ordinabili in funzione di dette etichette o tag di riordino (TID).

9. Procedimento secondo la rivendicazione 8, comprendente raccogliere in una coda, preferibilmente una coda FIFO, le etichette o tag di riordino (TID) accoppiate a transazioni cui è stato assegnato un identificatore consistente in uscita (CID) comune per tracciare dette transazioni e riordinarle al ritorno da detti moduli target (MEM1, MEM2).

10. Sistema per il trattamento di segnali comprendente una pluralità di moduli target (MEM1, MEM2) cui almeno un

modulo (10) accede tramite transazioni di accesso, il sistema, essendo configurato per operare con il procedimento secondo una qualsiasi delle rivendicazioni 1 a 9.

11. Sistema secondo la rivendicazione 10, in cui detto sistema è un System-on-Chip o SoC.

CLAIMS

1. A method of managing access transactions of at least one module (10) to a given target module (TRG) of a plurality of target modules (MEM1, MEM2) in a signal processing system, the method including assigning to said transactions respective transactions identifiers (ID, CID) and subjecting said transaction identifiers to a check (102, 104) by checking if an input identifier (ID) to the check (102, 104) has already been issued, wherein said input identifier (ID) to the check (102, 104) and said given target module (TRG) constitute an input identifier (ID)/given target module (TRG) pair to which a consistent output identifier (CID) is assigned, wherein:

- i) if said input identifier (ID) to the check (102, 104) has already been issued for a same given target module (TRG), assigning (106) said input identifier (ID) as a consistent output identifier (CID) to said input identifier (ID)/given target module (TRG) pair,

ii) if said input identifier (ID) to the check has not been already issued (102) or has been already issued (104) for a target module of said plurality (MEM1, MEM2) different from said given target module (TRG), assigning (108) to said input identifier (ID)/given target module (TRG) pair as a consistent output identifier (CID) a new identifier.

2. The method of claim 1, including generating (202) said input identifiers (ID) in disjoint ranges of identifiers, one range for each target module (TRG) in said plurality of target modules (MEM1, MEM2).

3. The method of claim 2, including generating (202) the input identifiers (ID) for a respective target module

(TRG) of said plurality of target modules (MEM1, MEM2) as the sum of a given identifier value and an offset $K \cdot \text{TRG}$, where K is a constant value and TRG is an integer identifying said respective target module of said plurality of target modules (MEM1, MEM2).

4. The method of any of claims 1 to 3, including:

- maintaining a table having as entries the input identifier/given target module (TRG) pairs for the current outstanding access transactions to said plurality of target modules (MEM1, MEM2), each pair having associated a count of the corresponding access transactions which are outstanding, wherein:

- i) if an input identifier (ID) to said check (102, 104) is contained in any entry of said table, assigning said input identifier (ID) as a consistent output identifier (CID) to said entry in the table and incrementing by one the associated count,

- ii) if said input identifier (ID) to said check is not contained in any entry of said table, filling an empty entry in said table and initializing a corresponding associated count for said empty entry being filled.

5. The method of claim 4, including checking if said table is full and, if the table is full, inhibiting further access transactions to said plurality of target modules (MEM1, MEM2),

6. the method of claim 4 or claim 5, including decrementing by one said associated count whenever a corresponding access transaction has been completed.

7. The method of any of the previous claims 4 to 6, including checking whether said associated count value is zero and, if zero, making the corresponding consistent identifier (CID) available to be assigned to a new

transaction.

8. The method of any of the previous claims, including:

- coupling to each said access transaction and the consistent output identifier (CID) assigned thereto a reordering tag (TID), and

- propagating towards said target modules (MEM1, MEM2) accessed by said transaction both said consistent output identifier (CID) assigned thereto and the reordering tag (TID) coupled therewith, whereby transactions returned from said target modules (MEM1, MEM2) are re-orderable as a function of said reordering tags (TID).

9. The method of claim 8, including collecting in a queue, preferably FIFO queue, the reordering tags (TID) coupled to transactions to which a common consistent transaction identifier (CID) has been assigned to permit tracking of said transactions and reordering them upon returning from said target module (MEM1, MEM2).

10. A signal processing system including a plurality of target modules (MEM1, MEM2) accessed by at least one module (10) via access transactions, the system configured to operate according to the method of any of claims 1 to 9.

11. The system of claim 10, wherein said system is a System-on-Chip or SoC.

FIG. 1

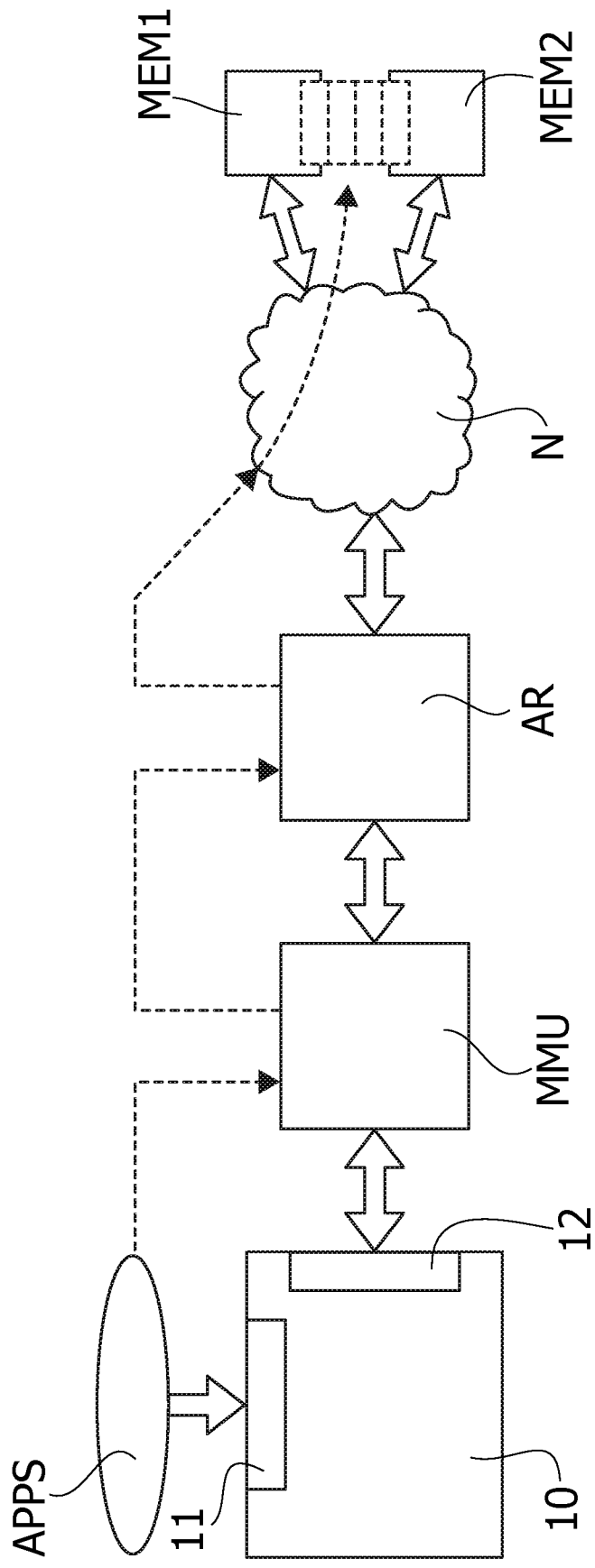
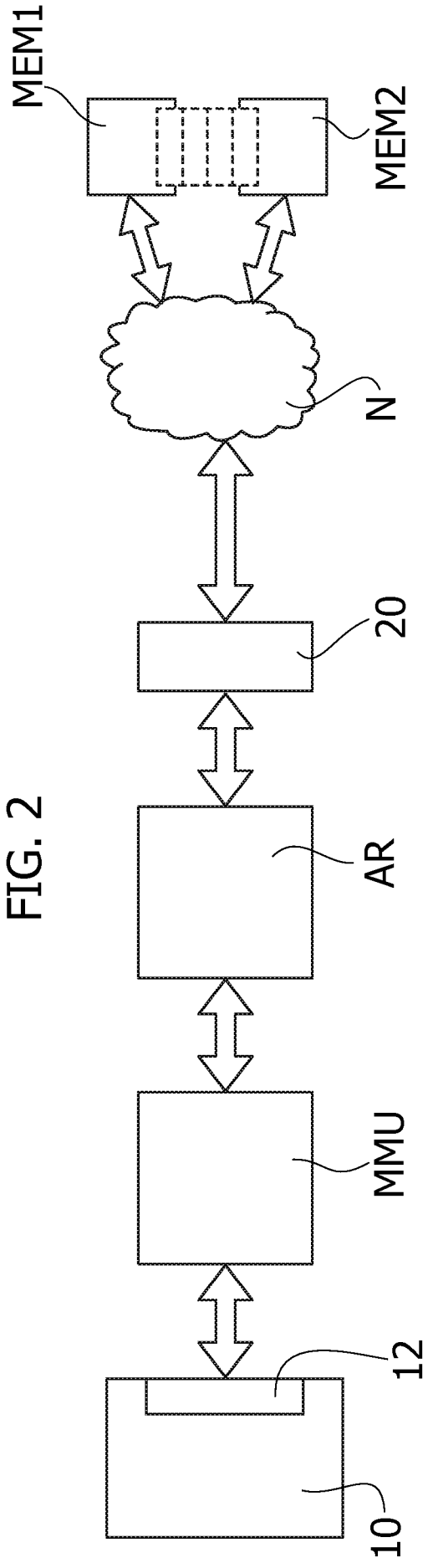


FIG. 2



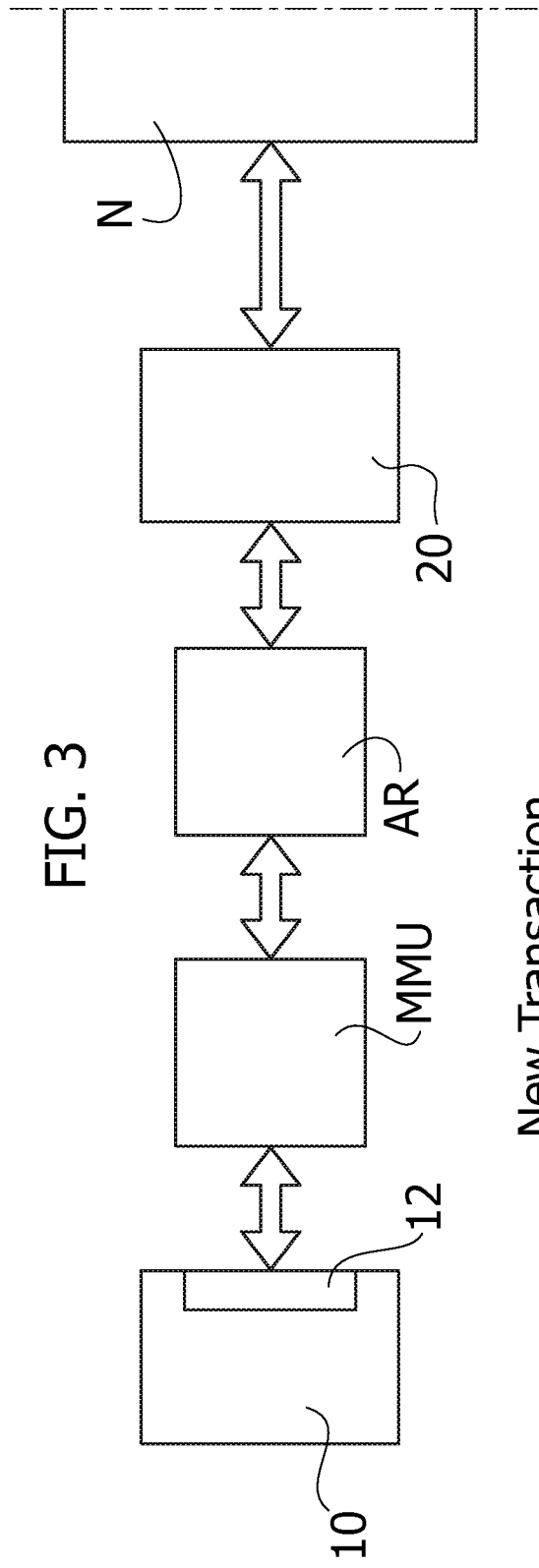


FIG. 4

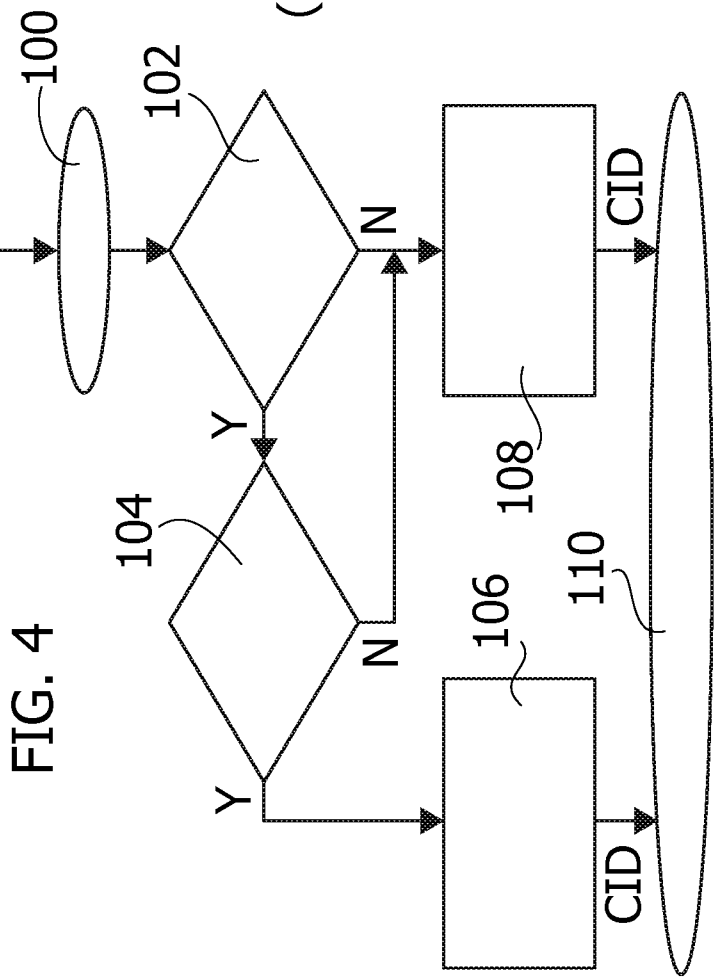


FIG. 7

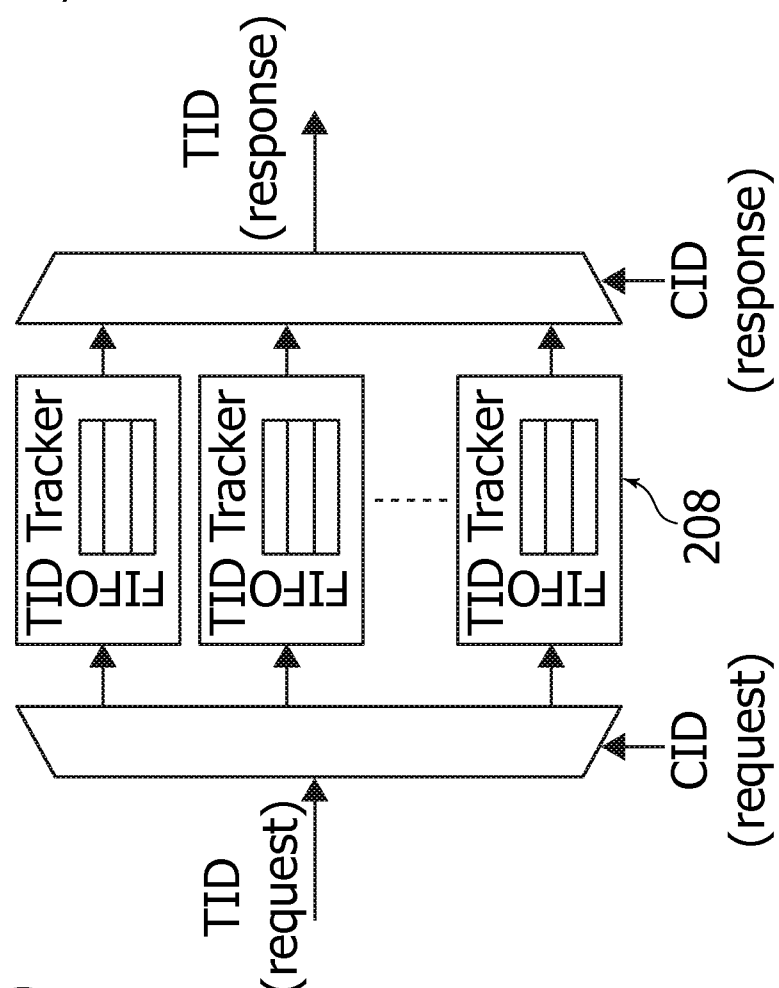


FIG. 5

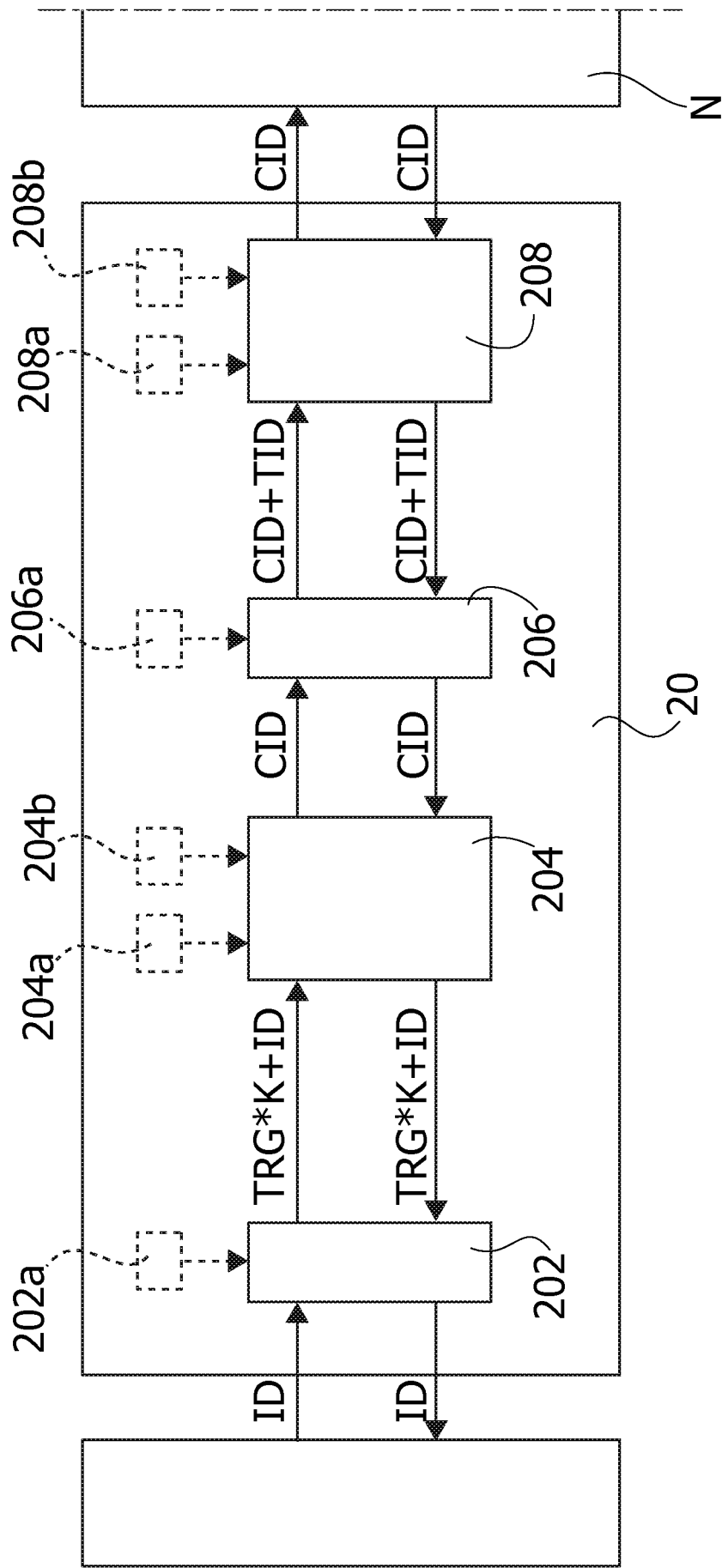


FIG. 6

