

ITALIAN PATENT OFFICE

Document No.

102012902048004A1

Publication Date

20131107

Applicant

STMICROELECTRONICS S.R.L.

Title

PROCEDIMENTO E DISPOSITIVO PER INTERROMPERE TRANSAZIONI,
SISTEMA E PRODOTTO INFORMATICO RELATIVI

DESCRIZIONE dell'invenzione industriale dal titolo:

"Procedimento e dispositivo per interrompere transazioni, sistema e prodotto informatico relativi"

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

Inventore designato: Daniele MANGANO

Depositata il: 7 maggio 2012

TESTO DELLA DESCRIZIONE

Campo tecnico

La descrizione si riferisce alle tecniche per interrompere (ossia "abortire", secondo una terminologia corrente) transazioni.

Varie forme di attuazione possono trovare applicazione in sistemi di trattamento di segnale.

Sfondo tecnologico

I sistemi complessi per il trattamento di segnali comprendono oggi giorno varie componenti hardware e software ciascuna delle quali ha il compito di assolvere a rispettive funzioni (ad esempio, visualizzazione di segnali video, funzioni grafiche, decodifica video, ecc.): si può trattare, ad esempio, di blocchi di logica e/o di dati indicati correntemente come nuclei o core IP (Intellectual Property).

In varie soluzioni, le componenti hardware e software del sistema possono essere interfacciate con le interconnessioni di sistema tramite un blocco di interfacciamento dedicato avente la funzione di generare le transazioni di bus.

In varie soluzioni, il blocco di interfacciamento può ricevere in ingresso le richieste delle varie componenti (ad esempio richieste di accesso a memoria - Direct Memory Access o DMA) e/o dei dati (ad esempio, nel caso di accessi in scrittura) ed è in grado di generare corrispondenti transazioni di bus con le caratteristiche richieste (indirizzi, formati, ecc.).

Lo schema a blocchi della figura 1 illustra, a puro titolo esemplificativo, un contesto del tipo in precedenza delineato. Nell'esempio della figura 1 si vede un modulo IP (un sistema complesso comprende in generale una pluralità di questi moduli) che, da un lato, si interfaccia con uno o più moduli esterni EXT e una CPU in funzione di host e, dall'altro lato, coopera con una struttura a bus (Bus Interface BI) tramite un blocco di interfacciamento PLUG. Ad esempio, il modulo IP può inviare verso il blocco PLUG comandi DMA Commands di accesso ad una memoria (ad esempio di tipo DDR, ossia Double Data Rate) e ricevere dal blocco PLUG dati, indicati con DATA, provenienti dalla memoria.

Lo schema della figura 1 fa vedere anche che il modulo IP può inviare verso il blocco di interfacciamento PLUG un comando, denominato Abort, destinato ad interrompere ("abortire") una o più transazioni in corso, in generale in numero pari a k , dove k può avere un qualsiasi valore. Secondo varie soluzioni, si può trattare di transazioni inerenti a funzioni di interconnessione (interconnect) e/o di controllo (lettura/scrittura) di una memoria (DDR CTRL).

In varie soluzioni (ed in varie forme di attuazione qui considerate), la funzionalità realizzata dal blocco di interfacciamento PLUG può essere di fatto integrata in un modulo IP: la distinzione qui fatta rileva quindi in via primaria dal punto di vista funzionale.

La figura 2 illustra, sempre a titolo schematico ed esemplificativo, il fatto che blocco di interfacciamento PLUG può avere un funzionamento bidirezionale con

- l'emissione di segnali relativi a transazioni di varia natura: Req (richiesta), TID (Transaction Identifier, ossia identificatore di transazione), DATA/Other (dati/informazioni di altra natura), EOT (End Of Transaction, ossia fine della transazione), e

- la ricezione di corrispondenti risposte RReq, RTID, RData/Other, REOT, che "ritornano" verso il blocco PLUG.

Il suddetto funzionamento bidirezionale può altresì comprendere la ricezione di un segnale di conferma Ack cui il blocco di interfacciamento PLUG dà riscontro con un corrispondente segnale di risposta RAck.

Nel commutare, ad esempio per effetto di un evento imprevisto, da un normale lavoro (task) NT in corso ad un altro lavoro, quale un task in tempo reale (Real Time) RTT, si desidera poter interrompere (ossia "abortire") quanto più rapidamente possibile il o i task in corso così da poter avviare l'esecuzione di uno più task nuovi.

I moderni sistemi di trattamento di segnale possono presentare un'elevata capacità di buffering (ad esempio a livello di rete integrata di calcolo - Integrated Computer Network o ICN - e/o di controllo di memoria): il numero di transazioni in corso da interrompere all'arrivo di un comando di Abort può quindi essere molto elevato.

Gli inventori hanno osservato che un lavoro o task di un nucleo IP può in realtà implicare più accessi a memoria (ad es. DMA), il che aumenta ancora il numero di transazioni su un bus come il bus BI della figura 2; ed ancora, anche se in linea di principio le transazioni associate ad uno stesso task possono essere eseguite una

per volta (il che implicherebbe di avere una sola transazione outstanding), nei sistemi moderni questa possibilità può essere preclusa dalla necessità delle IP di ottenere un throughput (o una banda) grande. Principalmente per questo motivo che i moderni protocolli e sistemi di comunicazione on-chip (come AMBA AXI o STBus) possono supportare transazioni multiple (outstanding), ovvero la possibilità di avere contemporaneamente diverse transazioni outstanding. In varie soluzioni, il numero di transazioni outstanding può dipendere da un certo numero di fattori, quali ad es. la capacità di buffering del sistema (Interconnect + Memory Controller), la particolare natura dell'IP, ecc.

Esiste quindi il problema di interrompere un numero anche elevato di transazioni in modo rapido e con costi contenuti: parlando di modo rapido si intende dire che, alla ricezione di un comando di Abort (emesso ad esempio tramite un segnale dedicato), si desidera mettere il nucleo o i vari nuclei IP in condizione di poter interrompere quanto prima possibile lo svolgimento dei task in corso così da essere in grado di assicurare lo svolgimento dei nuovi lavori o task, ad esempio in tempo reale.

Gli inventori hanno osservato che è possibile fare in modo che, dopo il segnale di interruzione (Abort), il blocco di interfacciamento vada in stallo, attendendo il "ritorno" (o "come-back") di tutte le transazioni in corso, mentre il funzionamento del blocco di interfacciamento si riavvia quando non ci sono più transazioni in corso.

Gli inventori hanno osservato che, in particolare in presenza di un numero elevato di transazioni pendenti, questa soluzione penalizza in modo significativo le prestazioni di sistema.

In taluni processori (vedere ad esempio il processore TI OMAP Multimedia, così come documentato al momento del deposito della presente domanda all'indirizzo web <http://www.ti.com/lit/ug/spru758A/spru758a.pdf>) per gestire gli errori di accesso si implementa una funzione di abort, senza però dare origine ad una soluzione in grado di gestire più transazioni pendenti.

Scopo e sintesi

Sussiste quindi l'esigenza di superare gli inconvenienti in precedenza delineati.

Secondo l'invenzione, tale scopo è raggiunto grazie ad un procedimento avente le caratteristiche richiamate nelle rivendicazioni che seguono. L'invenzione riguarda anche un corrispondente dispositivo, un corrispondente sistema nonché un prodotto informatico caricabile nella memoria di almeno un elaboratore e comprendente parti di codice software suscettibili di realizzare le fasi del procedimento quando il prodotto è eseguito su almeno un elaboratore. Così come qui utilizzato, il riferimento ad un tale prodotto informatico è inteso essere equivalente al riferimento ad un mezzo leggibile da elaboratore contenente istruzioni per il controllo del sistema di elaborazione per coordinare l'attuazione del procedimento secondo l'invenzione. Il riferimento ad "almeno ad un elaboratore" è evidentemente inteso a mettere in luce la possibilità che la presente invenzione sia attuata in forma modulare e/o distribuita.

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

Varie forme di attuazione possono basarsi sulla soluzione di trattare secondo uno schema di pipelining tanto lavori o task "vecchi" che vengono interrotti ("abortiti") quanto lavori o task "nuovi" di cui si avvia l'esecuzione.

In varie forme di attuazione, è possibile dare origine ad una soluzione efficiente che consente di trattare secondo uno schema di pipelining:

- da una parte, l'interruzione delle transazioni relative ad uno o più lavori "vecchi", ad es. con bassa priorità, e

- dall'altra parte, l'esecuzione di uno o più lavori "nuovi", ad es. in tempo reale.

Varie forme di attuazione possono essere applicate a moduli o nuclei (core) IP di qualunque natura (ad esempio CPU ovvero componenti hardware/software specializzati), ad esempio in relazione a qualsiasi protocollo di bus (STBus, AMBA AXI, ecc).

Secondo varie forme di attuazione, i dati e le transazioni di accesso a memoria (DMA) che all'arrivo del comando di interruzione (Abort) si trovano all'interno del blocco di interfacciamento possono essere scartati immediatamente.

Secondo varie forme di attuazione, le transazioni pendenti all'arrivo del comando di interruzione (Abort) possono essere marcate come interrotte ("abortite").

Varie forme di attuazione possono prevedere che le transazioni marcate come interrotte o abortite siano scartate tramite un'unità di filtro dedicata quando esse ritornano ("come back") al blocco di interfacciamento mentre questo sta già lavorando sui task nuovi.

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 un diagramma di flusso rappresentativo di forme di attuazione,
- la figura 4 è uno schema di forme di attuazione,
- la figura 5 è una rappresentazione esemplificativa del funzionamento di forme di attuazione,
- la figura 6 è una rappresentazione schematica di forme di attuazione,
- le figure 7 e 8 sono diagrammi di stato relativi a forme di attuazione e,
- la figura 9 illustra una possibile implementazione 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.

Le figure 3 a 9 fanno riferimento a titolo di esempio ad un contesto di sistema di trattamento del tipo di quello già in precedenza esemplificato con riferimento alle figure 1 a 2. La descrizione già fornita in relazione a tali figure si applica quindi anche alle varie forme di attuazione nel seguito esemplificate, se non per quanto nel seguito diversamente descritto. Questo vale, solo per fare un esempio, per quanto riguarda la possibilità di integrare in un modulo IP la funzionalità dal blocco di interfacciamento PLUG.

Le figure 3 e 4 forniscono una prima rappresentazione di possibili forme di attuazione relative alle modalità di funzionamento attuabili alla ricezione di un segnale (indicato con Abort) che indica, ad esempio al modulo di interfacciamento PLUG rappresentato nella figura 1, di interrompere uno o più lavori (task) in corso al fine di poter dare corso ad uno o più nuovi lavori, ad esempio lavori che assumono priorità rispetto ai precedenti, ad esempio perché si tratta di lavori da svolgere in tempo reale (Real Time Task o RTT).

In varie forme di attuazione, alla ricezione del segnale di Abort, il blocco di interfacciamento PLUG può

quindi avviare, in un passo 100, la corrispondente procedura e provvedere, in un successivo passo 102, a scartare i dati e/o le richieste di accesso (DMA) pendenti che si trovano all'interno del blocco di interfacciamento PLUG stesso.

In varie forme di attuazione, in un successivo passo 104, il blocco di interfacciamento PLUG può marcare il o i segnali identificatori di transazione (ad es. i cosiddetti Transaction Identifier or TID) relativi alle transazioni in corso come interrotti o "abortiti" ed avviare (secondo un generale schema di pipelining) lo svolgimento del nuovo o dei nuovi lavori o task. In varie forme di attuazione, quando il blocco di interfacciamento PLUG riavvia il funzionamento normale e comincia a trattare il nuovo o i nuovi lavori, come relativi identificatori di transazione "nuovi" si possono utilizzare (anche) identificatori precedentemente associati a transazioni "abortite" che sono state eliminate al loro ritorno.

In varie forme di attuazione, i passi indicati con 100, 102 e 104 possono quindi essere visti come rappresentativi di una funzione interna del blocco di interfacciamento PLUG.

I passi successivi del diagramma di flusso della figura 3 si impernano su un passo 106 suscettibile di essere visto come un passo di attesa del "ritorno" (come back) delle transazione in corso.

In varie forme di attuazione, a partire dal passo 106, in un passo 108 il blocco di interfacciamento PLUG può verificare se gli identificatori delle transazioni di ritorno sono fra quelli che sono stati marcati come "abortiti" nel passo 104 (dunque corrispondenti a

transazioni relative a lavori "vecchi" da interrompere) oppure no.

Se il passo 108 dà esito negativo (N, il che indica trattarsi di una transazione relativa ad un lavoro "nuovo", da non interrompere, il cui identificatore non è marcato come "abortito") si ritorna semplicemente al passo 106.

Se invece il passo 108 dà esito positivo (Y, il che indica trattarsi di una transazione relativa ad un lavoro "vecchio", da interrompere, il cui identificatore è stato marcato come "abortito"), in varie forme di attuazione, il blocco di interfacciamento PLUG può provvedere in un successivo passo 110 ad eliminare la relativa transazione. In varie forme di attuazione, il blocco di interfacciamento PLUG può rendere l'identificatore corrispondente nuovamente disponibile per l'assegnazione ad una (nuova) transazione.

Il riferimento 112 indica un passo destinato a verificare se ci sono ancora identificatori marcati come "abortiti" (ossia a verificare se ci sono ancora transazioni di cui si attende il ritorno).

Se il passo 112 dà esito positivo (Y), si ritorna al passo 106. Se il passo 112 dà esito negativo (N), la procedura può dirsi completata in un passo 114.

In varie forme di attuazione, i passi 106 a 114 possono essere visti come rappresentativi della funzione dedicata di filtraggio di cui si è detto in precedenza.

Esaminata in diretto confronto con la figura 2, la figura 4 evidenzia il fatto che i criteri di funzionamento illustrati a titolo di esempio in relazione al diagramma di flusso della figura 3 possono essere visti come espressione della possibilità di prevedere, accoppiati o in associazione al blocco di interfacciamento PLUG:

- un modulo 120 suscettibile di realizzare una funzione di verifica (abortito/non abortito) sugli identificatori, ad esempio a partire dal relativo segnale di risposta RTID, generando un corrispondente segnale EN, e
- un modulo 122 suscettibile di svolgere la suddetta funzione di filtraggio in funzione valore del segnale EN, scartando, ad esempio a livello del segnale di risposta RReq, le transazioni relative ai lavori o task "vecchi" che vengono interrotti; in varie forme di attuazione, questo può avvenire rinviando verso il blocco di interfacciamento PLUG le sole risposte RReq' corrispondenti ai lavori o task "nuovi".

Parlando di moduli 120, 122 "accoppiati o in associazione" al blocco di interfacciamento PLUG si intende evidenziare il fatto che tali moduli non devono di necessità essere interni al blocco PLUG: ad esempio, nella figura 4, il modulo 120 è illustrato all'interno ed il modulo 122 è all'esterno.

Lo schema della figura 5 illustra a livello di schema sinottico generale una possibile forma di attuazione dei criteri di funzionamento presentati a titolo esemplificativo in precedenza.

In particolare, la schema della figura 5 può essere visto come idealmente ripartito in due aree o zone relative rispettivamente:

- alla esecuzione, prima della ricezione del segnale di Abort, di transazioni T11 a T19 relative a lavori o task "vecchi" o "normali", destinati ad essere interrotti a seguito della ricezione del segnale di Abort stesso (area o zona NT, a sinistra), e
- all'avviamento, alla ricezione del segnale di Abort, di transazioni T21 a T28 relative a lavori o task "nuovi"

T21 a T28, ad esempio in tempo reale, la cui esecuzione è avviata in pipelining alla eliminazione delle transazioni relative ai lavori vecchi (area o zona RTT, a destra).

Le caselle barrate corrispondono appunto a:

- dati e/o transazioni quali richieste di accesso a memoria (T12, T13, T17, T18, T19) che, all'arrivo del segnale di Abort, si trovano all'interno del blocco di interfacciamento PLUG e vengono scartati (DISCARD),

- transazioni (T14, T15, T16) che, all'arrivo del segnale di Abort, sono pendenti e vengono eliminate al loro ritorno secondo il meccanismo di filtraggio (FILTERING) descritto in precedenza, attuato mentre è già in corso l'esecuzione delle transazioni relative ai lavori nuovi.

Il relazione alle caselle T11, T13, T16 nella colonna identificata dal simbolo del segnale di Abort e non barrate, l'esempio qui presentato - che è tale - si basa sulle seguenti assunzioni:

- nell'istante in cui arriva il segnale di Abort, si assume che la transazione T11 sia stata già letta dall'IP: l'istante precedente, ovvero quello corrispondente alla colonna di sinistra, la stessa transazione si trovava sul bus di risposta ed era stata caricata nella memoria interna del blocco PLUG;

- la transazione T13 si assume stare rientrando esattamente nel momento in cui arriva il segnale di Abort. Tale transazione viene caricata contestualmente nella memoria interna del PLUG e abortita (la stessa T13 si trova nella riga sotto ed è barrata): nell'esempio di figura 5 si assume dunque che una transazione di risposta sul bus venga caricata immediatamente nel modulo PLUG (in corrispondenza della stessa colonna). In varie forme di attuazione si potrebbe assumere per esempio che una transazione di

risposta sul bus venga caricata nel PLUG successivamente (colonna successiva dopo);

- all'arrivo del segnale di Abort si assume che la transazione T16 sia stata già inviata dal blocco PLUG sul bus (esattamente nello stesso istante). Di conseguenza, tale transazione è marcata internamente al blocco PLUG come aborted e quindi la corrispondente transazione di risposta verrà filtrata al momento opportuno secondo lo schema descritto precedentemente: in particolare, si può osservare ancora una volta in figura 5 che la transazione T16 si assume essere ricevuta sul bus di risposta e filtrata quattro colonne dopo.

Si apprezzerà altresì che, per semplicità di trattazione, la figura 5 non contiene alcun riferimento agli identificatori di transazioni. I nomi assegnati alle transazioni hanno il solo scopo di rendere evidente l'associazione tra task e transazione: in particolare, il primo indice identifica il task, il secondo indice identifica la transazione all'interno dello stesso task.

Il diagramma della figura 5 può essere quindi visto come costituito da cinque righe sovrapposte, ciascuna relativa ad uno stadio della pipeline, ossia:

- Riga 1: transazioni di richiesta memorizzate nel blocco PLUG (Request transactions stored in the PLUG)
- Riga 2: transazioni di richiesta bus (Bus request transactions)
- Riga 3: transazioni di risposta bus (Bus response transactions)
- Riga 4: transazioni di risposta memorizzate nel blocco PLUG (Response transactions stored in the PLUG)
- Riga 5: transazioni di risposta ricevute dal modulo IP (Response transactions received by the IP).

In varie forme di attuazione, la procedura qui esemplificata permette di mantenere piena la pipeline anche dopo il comando di Abort, il che corrisponde a modalità di utilizzazione efficienti delle risorse lavorative.

In varie forme di attuazione, è possibile dare origine ad una soluzione del tutto indipendente dal protocollo o dai protocolli utilizzati. In particolare, nel caso di protocolli che non comprendano segnali identificatori di tipo TID, è possibile far ricorso ad identificatori equivalenti, se del caso utilizzando corrispondenti regole di ordinamento: un possibile esempio è dato dagli identificatori noti in contesto AMBA come identificatori AXI Source ID.

In varie forme di attuazione, è anche possibile procedere ad un interleaving degli identificatori "abortiti" e "non- abortiti".

Varie forme di attuazione permettono di trattare anche il caso in cui, prima del completamento di una procedura così come esemplificata (ossia prima di arrivare alla fase 114), arriva un altro segnale di Abort: varie forme di attuazione permettono infatti di ripartire da capo quale che sia lo stato corrente del sistema.

In termini di prestazioni, gli esempi di attuazione qui considerati permettono di operare una procedura di interruzione (abort) con estrema rapidità, ad esempio in un solo ciclo di clock.

Tutti gli identificatori (TID o analoghi) non abortiti possono essere utilizzati immediatamente per un nuovo task (con un contenimento delle dimensioni) e, quando un identificatore già marcato come abortito ritorna, lo stesso può essere reso immediatamente disponibile per un nuovo task.

Per quanto riguarda il supporto a livello hardware, gli stati interni degli identificatori possono essere aggiornati per aggiungere identificatori abortiti (marcatura delle transazioni abortite) mentre la funzione di filtro (ad esempio, il blocco 122 nella figura 4) può essere gestito con una logica di controllo comune che si incarica anche di gestire gli stati degli identificatori.

In varie forme di attuazione, questa funzione, ossia la funzione di marcatura delle transazioni destinate ad essere abortite, può essere svolta tramite un meccanismo che rappresenta lo stato degli identificatori a livello vettoriale (ad es. TID Vector State o TVS, sempre rifacendosi all'esempio degli identificatori TID) così come schematicamente rappresentato nello schema della figura 6 dove il vettore indicato con TVS rappresenta la sequenza degli stati S1, S2, ..., Sn dei vari identificatori TID1, TID2, ..., TIDn.

In varie forme di attuazione, il vettore TVS può essere utilizzato nel modulo di interfacciamento PLUG per svolgere anche funzioni come il riordino, la rilevazione della fine degli accessi (DMA), ecc.

Lo schema della figura 7 è un diagramma di stato comprendente rispettivamente due stati 200 e 202 che possono essere visti come:

- uno stato "valido": l'identificatore (ad es. TID) è disponibile e può essere assegnato ad una nuova transazione, e
- uno stato "non valido": l'identificatore (ad es. TID) è relativo ad una transazione pendente e non può essere utilizzato.

In generale, nel funzionamento normale, così come rappresentato nella figura 7, il passaggio dallo stato 200

allo stato 202 si ha quando un identificatore viene assegnato (TID assigned), mentre il passaggio in verso opposto, dallo stato 202 allo stato 200, si ha quando la relativa transazione "ritorna" (TID return).

Lo schema della figura 8 illustra una possibile forma di applicazione del meccanismo descritto in precedenza al diagramma di stato della figura 7.

Il diagramma della figura 8 prevede la presenza di un ulteriore stato 204 rappresentativo del fatto che un certo identificatore (ad es. TID) assegnato ad una transazione pendente che deve essere abortita è marcato come "abortito", ad esempio nel passo 104 della figura 3, a seguito della ricezione del segnale di Abort.

In quanto marcato come abortito, l'identificatore in questione passa dallo stato 202 allo stato 204 e, quando la relativa transazione viene interrotta, l'identificatore passa dallo stato 204 allo stato 200, ritornando ad essere a tutti gli effetti un identificatore "valido" assegnabile ad una (nuova) transazione.

Lo schema della figura 9 illustra, sempre a titolo esemplificativo, possibili forme di implementazione di maggior dettaglio dei blocchi 120 e 122 già presentati con riferimento alla figura 4. Il tutto mettendo in luce il fatto che le relative funzionalità possono prescindere dalla localizzazione all'interno o all'esterno del blocco PLUG.

Nella figura 9 il riferimento 124 indica la circuitazione logica del blocco di interfacciamento PLUG, suscettibile di comprendere, ad esempio, le funzioni di gestione delle code dei dati (ad esempio secondo schemi FIFO), gestione degli accessi (DMA) pendenti, invio e

ricezione degli altri segnali di bus (vedere al riguardo lo schema generale della figura 2).

Così come già detto in precedenza con riferimento alla figura 4, in varie forme di attuazione il blocco di filtraggio 122 può operare rinviando verso la logica 124, come risposte RReq', le risposte RReq che emergono come relative a transazioni "nuove" (da non abortire, ma da eseguire) a seguito dalla funzione di filtraggio svolta in funzione del segnale EN proveniente dal blocco 120 indicativo del fatto che il relativo identificatore sia marcato come abortito oppure no.

Adottando il formalismo introdotto con riferimento alle figure 6 a 8, il segnale EN può essere visto come proveniente da un multiplexer 126 posto all'uscita di un banco di registri che implementa fisicamente il vettore TVS con ciascun registro destinato a memorizzare lo stato S1, S2, ..., Sn dei vari identificatori. Il banco di registri TVS ed il multiplexer 126 cooperano con un modulo di controllo 128.

Nell'esempio di attuazione qui considerato, il blocco 128 è sensibile ai seguenti ingressi:

- segnale Abort (utilizzabile anche come segnale di reset RST verso la logica 124),
- un segnale RTID rappresentativo del ritorno dei singoli identificatori (ad es. TID), e
- il segnale REOT che veicola la risposta relativa alla fine della transazione.

Si apprezzerà che i segnali in questione possono essere già presenti in soluzioni come quelle cui fanno riferimento le figure 1 e 2.

In varie forme di attuazione (secondo le specifiche modalità di implementazione del blocco di interfacciamento

PLUG e dei relativi protocolli sono naturalmente possibili soluzioni diverse), quando viene asserito il segnale Abort, il blocco 128 aggiorna - ponendoli allo stato "abortito" (204 nella figura 8) - tutti gli identificatori "non validi" (stato 202 delle figure 7 e 8) in quanto al momento assegnati a transazioni pendenti.

In varie forme di attuazione questo può avvenire, così come schematicamente indicato in 130, in funzione di segnali generati dalla logica 124 al fine di gestire il normale funzionamento del registro TVS, ossia per quanto riguarda le transizioni fra gli stati "valido" 200 e "non valido" 202 così come rappresentati nella figura 7. Tutto questo così da permettere, ad esempio, quando si rivela una transazione come terminata (REOT=1), di far passare nuovamente allo stato "valido" 200 lo stato dell'identificatore compreso nel vettore TVS corrispondente a tale transazione.

In funzione di questo funzionamento, il multiplexer 126 permette in varie forme di attuazione di selezionare lo stato dello specifico identificatore (ossia dello specifico elemento del vettore TVS) associato ad una data transazione che "ritorna" (identificata dal segnale RTID di volta in volta coinvolto) così da poter attribuire un corrispondente valore al segnale EN.

In varie forme di attuazione, il segnale EN può assumere due stati o livelli:

- un primo livello (ad es. filter_EN=1), tale da attribuire al segnale RReq' trasferito verso la logica 124 un valore "non asserito", ossia pari a zero, così da far sì che la rispettiva transazione (relativa ad un task "vecchio", da interrompere) sia abortita, e

- un secondo livello (ad es. filter_EN=0), tale da attribuire al segnale RReq' trasferito verso la logica 124 un valore "asserito", ad es. RReq'=RReq, tale da far sì che la rispettiva transazione (relativa ad un task "nuovo") non sia abortita ma eseguita.

Varie forme di attuazione consentono quindi di accelerare notevolmente la funzione di interruzione (Abort) delle transazioni tramite una funzione di marcatura e di filtraggio delle transazioni da abortire.

La relativa procedura può essere utilizzata per attuare un'interruzione rapida delle transazioni in corso in un sistema di trattamento ricorrendo ad una microarchitettura generale nel complesso semplice (ad es. vettore degli stati TVS e relativi circuiti associati e unità di filtraggio 122).

Varie forme di attuazione possono migliorare la gestione dei lavori o task in tempo reale superando gli inconvenienti relativi alla possibile presenza di un numero elevato di transazioni pendenti (situazione tipica dei sistemi moderni con esteso ricorso a strutture di buffering)

Varie forme di attuazione possono comportare vantaggi in termini di miglioramento delle prestazioni di sistema, in particolare per quanto riguarda la gestione in tempo reale, con miglioramenti a livello di prodotti in termini di flessibilità e versatilità. Il tutto ricorrendo a soluzioni utilizzabili per un'ampia gamma di moduli IP e di sistemi (comprese CPU), con soluzioni indipendenti dagli specifici protocolli utilizzati.

Naturalmente, fermo restando il principio dell'invenzione, i particolari di realizzazione e 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 dell'invenzione; tale ambito di protezione è definito dalle rivendicazioni.

RIVENDICAZIONI

1. Procedimento per interrompere lavori o task correnti (T14, T15, T16) in esecuzione su un insieme di moduli (IP) di un sistema per il trattamento di segnali gestiti tramite un blocco di interfacciamento (PLUG) al fine di consentire l'esecuzione di nuovi lavori o task (T21 - T28) in detto insieme di moduli (IP), il procedimento comprendendo interrompere detti lavori o task correnti eliminando transazioni (T14, T15, T16) di detti lavori o task correnti in pipelining con l'esecuzione di transazioni (T21 - T28) di detti nuovi lavori o task.

2. Procedimento secondo la rivendicazione 1, comprendente, all'arrivo di un segnale (Abort) di interrompere detti lavori o task correnti, scartare (102) dati e/o accessi di memoria (T12, T13, T17, T18, T19) presenti in detto blocco di interfacciamento (PLUG).

3. Procedimento secondo la rivendicazione 1 o la rivendicazione 2, comprendente, a seguito dell'arrivo di un segnale (Abort) di interrompere detti lavori o task correnti:

- marcare (104) come interrotte le transazioni pendenti di detti lavori o task correnti, e
- eliminare (110) le transazioni marcate come interrotte (104) quando esse ritornano a detto blocco di interfacciamento (PLUG) durante l'esecuzione di dette transazioni (T21 - T28) di detti nuovi lavori o task.

4. Procedimento secondo la rivendicazione 3, comprendente:

- assegnare a dette transazioni rispettivi identificatori di transazione,

- all'arrivo di un segnale (Abort) di interrompere detti lavori o task correnti, marcare (104) come interrotti o abortiti (204) gli identificatori assegnati alle transazioni pendenti di detti lavori o task correnti, e

- eliminare (110, 122) le transazioni i cui identificatori sono marcati come interrotti o abortiti (104) filtrandole dalle transazioni di ritorno a detto blocco di interfacciamento (PLUG) durante l'esecuzione di dette transazioni (T21 - T28) di detti nuovi lavori o task.

5. Procedimento secondo una qualsiasi delle precedenti rivendicazioni, in cui a dette transazioni sono assegnati rispettivi identificatori di transazione, il procedimento comprendendo assegnare a transazioni (T21 - T28) di detti nuovi lavori o task identificatori di transazione precedentemente assegnati a transazioni di detti lavori o task correnti che sono state eliminate.

6. Dispositivo per interrompere lavori o task correnti (T14, T15, T16) in esecuzione su un insieme di moduli (IP) di un sistema per il trattamento di segnali gestiti tramite un blocco di interfacciamento (PLUG) al fine di consentire l'esecuzione di nuovi lavori o task (T21 - T28) in detto insieme di moduli (IP), il dispositivo comprendendo almeno un modulo (122, 126, 128, TVS) accoppiato a detto blocco di interfacciamento (PLUG) e configurato per interrompere detti lavori o task correnti eliminando transazioni (T14, T15, T16) di detti lavori o task correnti in pipelining con l'esecuzione di transazioni (T21 - T28) di detti nuovi lavori o task.

7. Dispositivo secondo la rivendicazione 6, il dispositivo (122, 126, 128, TVS) essendo sensibile (128) ad un segnale (Abort) di interrompere detti lavori o task correnti e configurato per scartare (102) dati e/o accessi di memoria (T12, T13, T17, T18, T19) presenti in detto blocco di interfacciamento (PLUG) all'arrivo di detto segnale (Abort) di interrompere detti lavori o task correnti.

8. Dispositivo secondo la rivendicazione 6 o la rivendicazione 7, il dispositivo (122, 126, 128, TVS) comprendendo:

- un modulo di controllo (128) sensibile (128) ad un segnale (Abort) di interrompere detti lavori o task correnti comprendente,

- un modulo di memoria (TVS) per marcare come interrotte, all'arrivo di detto segnale (Abort) di interrompere detti lavori o task correnti, le transazioni pendenti di detti lavori o task correnti, e

- un modulo di filtraggio (122, 126) per eliminare le transazioni marcate come interrotte quando esse ritornano a detto blocco di interfacciamento (PLUG) durante l'esecuzione di dette transazioni (T21 - T28) di detti nuovi lavori o task.

9. Dispositivo secondo la rivendicazione 8, in cui:

- detto modulo di memoria (TVS) è configurato per memorizzare lo stato (200, 202, 204) di rispettivi identificatori di transazione (S1, S2, ..., Sn) assegnati a dette transazioni,

- detto modulo di controllo (128) è configurato per attribuire, all'arrivo di un segnale (Abort) di interrompere detti lavori o task correnti, uno stato di interrotto o abortito (204) agli identificati assegnati alle transazioni pendenti di detti lavori o task correnti, e

- detto modulo di filtraggio (122, 126) è configurato per eliminare le transazioni i cui identificatori hanno assegnato uno stato di interrotto o abortito (204) filtrandole dalle transazioni di ritorno a detto blocco di interfacciamento (PLUG) durante l'esecuzione di dette transazioni (T21 - T28) di detti nuovi lavori o task.

10. Dispositivo secondo una qualsiasi delle precedenti rivendicazioni 6 a 9, in cui a dette transazioni sono assegnati rispettivi identificatori di transazione, il dispositivo essendo configurato per assegnare a transazioni (T21 - T28) di detti nuovi lavori o task identificatori di transazione precedentemente assegnati a transazioni di detti lavori o task correnti che sono state eliminate.

11. Sistema per il trattamento di segnali comprendente un insieme di moduli (IP) gestiti tramite un blocco di interfacciamento (PLUG), il sistema comprendendo un dispositivo secondo una delle rivendicazioni 6 a 10 per interrompere lavori o task correnti (T14, T15, T16) in esecuzione al fine di consentire l'esecuzione di nuovi lavori o task (T21 - T28) in detto insieme di moduli (IP).

12. Prodotto informatico, caricabile nella memoria di almeno un elaboratore e comprendente porzioni di codice

software per implementare tutti i passi di un procedimento secondo una qualsiasi delle rivendicazioni 1 a 5.

CLAIMS

1. A method of aborting current tasks (T14, T15, T16) being executed in a set of modules (IP) of a signal processing system managed via an interface block (PLUG) in order to permit executing new tasks (T21 - T28) in said set of modules (IP), the method including aborting said current tasks by pipelining eliminating transactions (T14, T15, T16) of said current tasks and executing transactions (T21 - T28) of said new tasks.

2. The method of claim 1, including, upon arrival of a signal (Abort) to abort said current tasks, discarding (102) data and/or memory accesses (T12, T13, T17, T18, T19) present in said interface block (PLUG).

3. The method of claim 1 or claim 2, including, as a result of the arrival of a signal (Abort) to abort said current tasks:

- marking (104) pending transactions of current tasks as aborted, and
- eliminating (110) transactions marked as aborted (104) as they return to said interface block (PLUG) during the execution of transactions (T21 - T28) of said new tasks.

4. The method of claim 3, including:

- assigning respective identifiers to said transactions,
- upon arrival of a signal (Abort) to abort said current tasks, marking (104) as aborted (204) the

identifiers assigned to the pending transactions of said current tasks, and

- eliminating (110, 122) transactions whose identifiers are marked as aborted (104) by filtering them out of transaction coming back to said interface block (PLUG) during the execution of such transactions (T21 - T28) of said new tasks.

5. The method of any of the preceding claims, wherein said transactions are assigned respective transaction identifiers, the method including assigning to transactions (T21 - T28) of said new tasks transaction identifiers previously assigned to transactions of said current tasks that have been eliminated.

6. A device for aborting current tasks (T14, T15, T16) being executed in a set of modules (IP) of a signal processing system managed via an interface block (PLUG) in order to permit executing new tasks (T21 - T28) in said set of modules (IP), the device including at least one module (122, 126, 128, TVS) coupled to said interface block (PLUG) and configured for aborting said current tasks by pipelining eliminating transactions (T14, T15, T16) of said current tasks and executing transactions (T21 - T28) of said new tasks.

7. The device of claim 6, the device (122, 126, 128, TVS) sensitive (128) to a signal (Abort) to abort said current tasks and configured to discard (102) data and/or memory accesses (T12, T13, T17, T18, T19) present in said interface block (PLUG) upon arrival of said signal (Abort) to abort said current tasks.

8. The device of claim 6 or claim 7, the device (122, 126, 128, TVS) including:

- a control module (128) sensitive (128) to a signal (Abort) to abort said current tasks,
- a memory module (TVS) to mark as aborted pending transactions of said current tasks upon arrival of said signal (Abort) to abort said current tasks, and
- a filtering module (122, 126) to eliminate transactions marked as aborted as they come back to the interface block (PLUG) during execution of said transactions (T21 - T28) of said new tasks.

9. The device of claim 8, wherein:

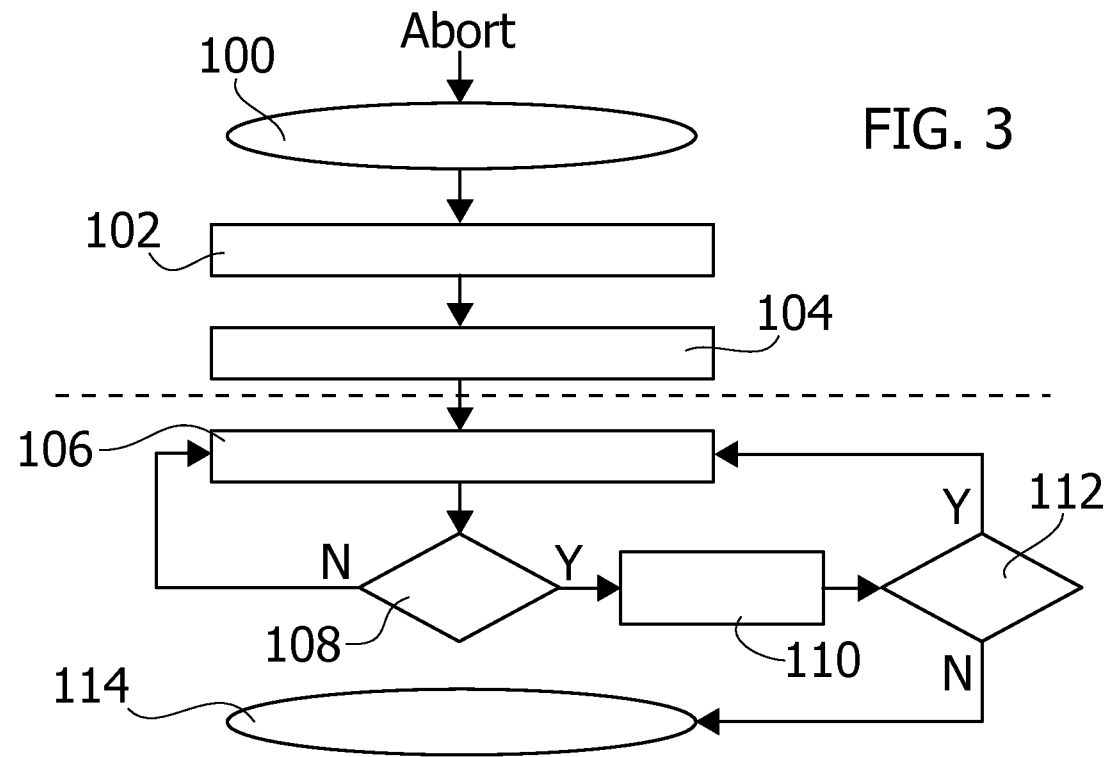
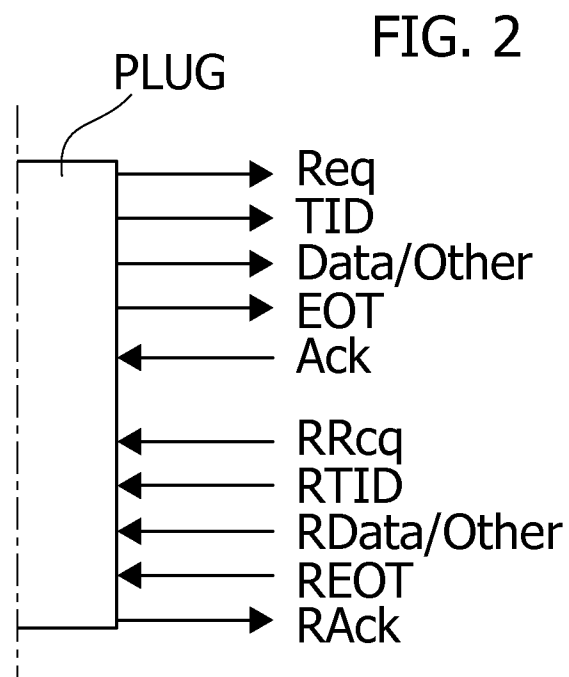
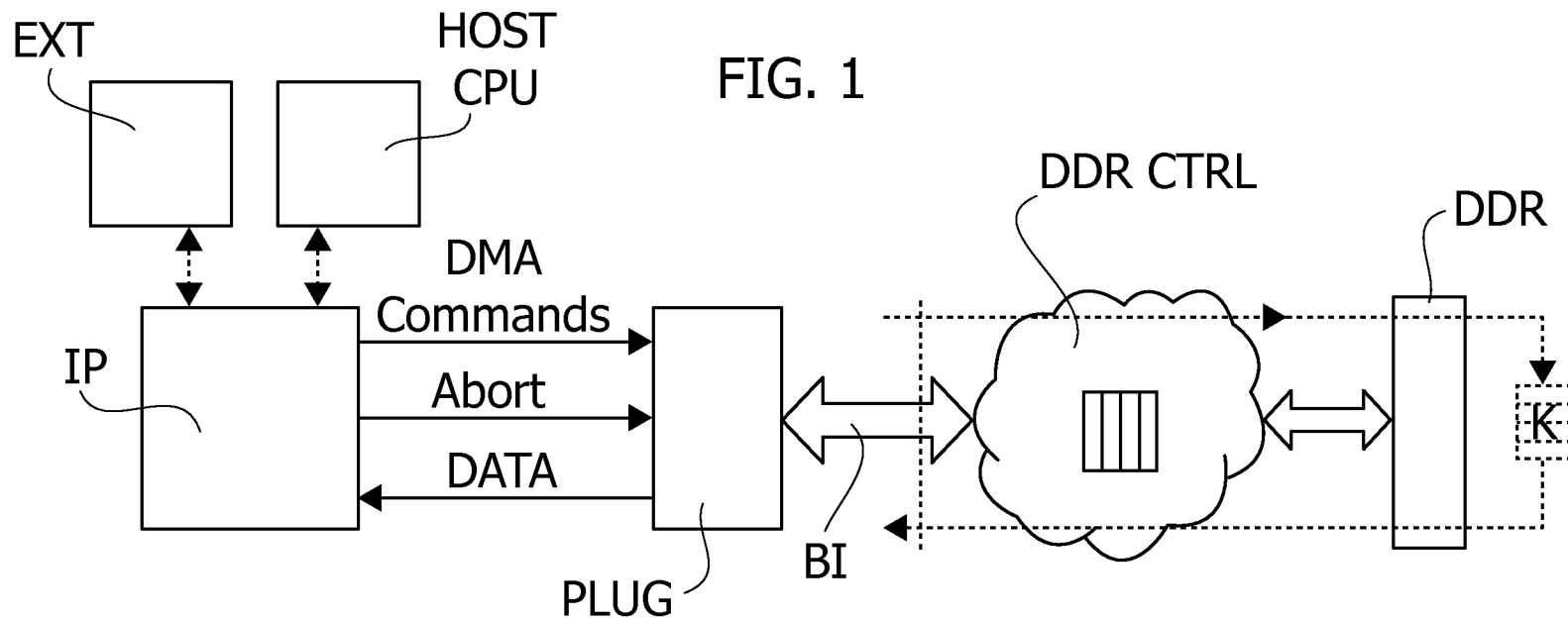
- said memory module (TVS) is configured for storing the state (200, 202, 204) of the respective transaction identifiers (S1, S2, ..., Sn) assigned to said transactions of said current tasks,
- said control module (128) is configured for assigning, upon arrival of a signal (Abort) to abort said current tasks, an aborted state (204) to identifiers assigned to pending transactions of said current tasks, and
- said filter module (122, 126) is configured for eliminating the transactions whose identifiers are assigned an aborted state (204) by filtering them out of the transactions coming back to said interface block (PLUG) during the execution of said transactions (T21 - T28) of said new tasks.

10. The device of any of the preceding claims 6 to 9, wherein said transactions are assigned respective identifiers of transaction, the device configured for assigning to transactions (T21 - T28) of said new tasks

transaction identifier previously assigned to transactions of said current tasks that have been eliminated.

11. A signal processing system including a set of modules (IP) managed via an interface block (PLUG), the system including a device according to one of claims 6 to 10 to abort current tasks (T14, T15, T16) being executed in order to permit executing new tasks (T21 - T28) in said set of modules (IP).

12. A computer program product, loadable in the memory of at least one computer and including software code portions to implement all the steps of a method according to any one of claims 1 to 5.



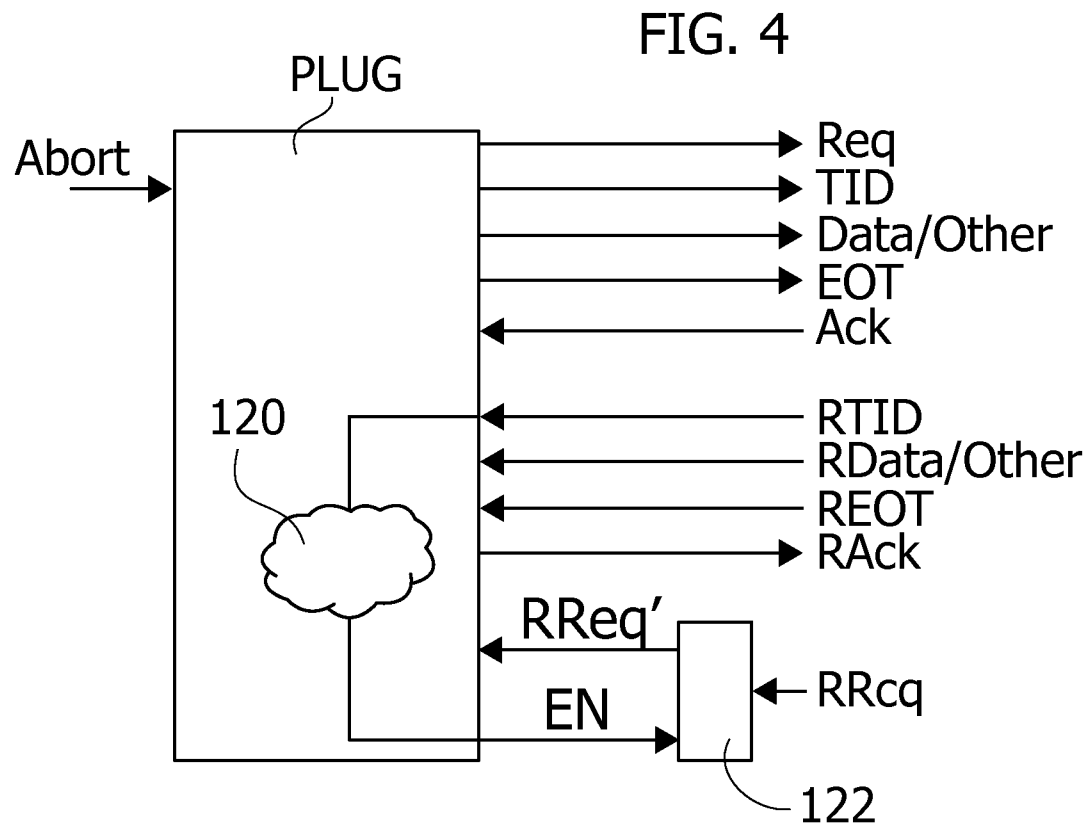
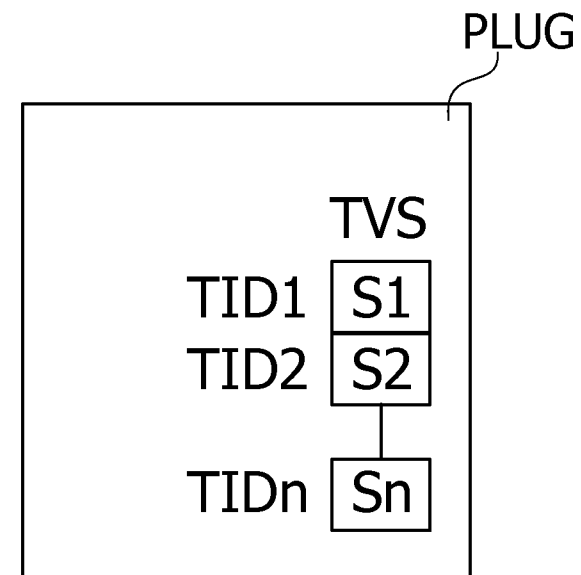


FIG. 6



2/4

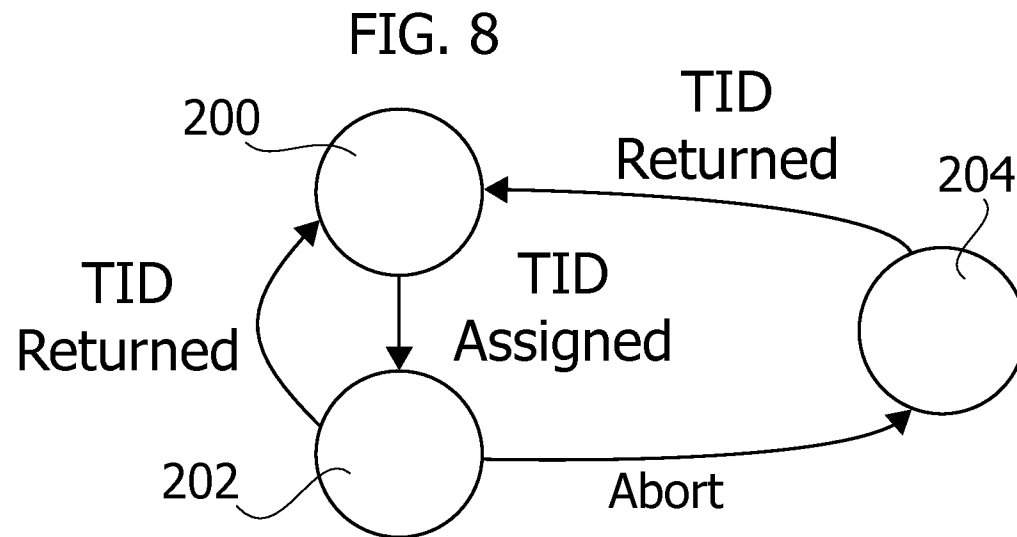
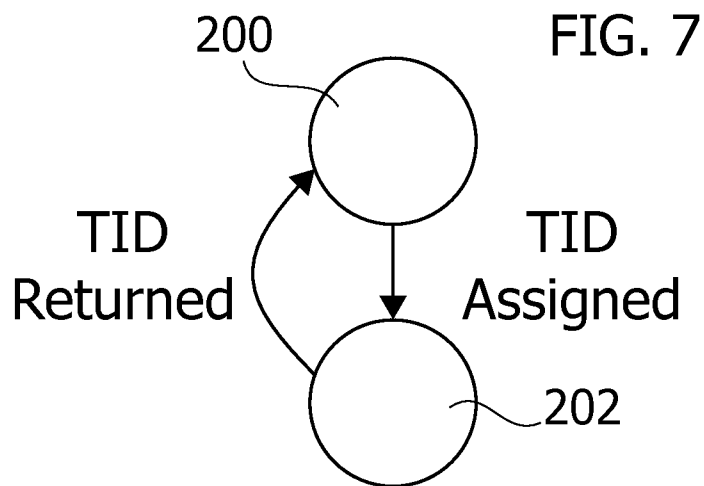




FIG. 5

NT

RTT

DISCARD

T14	T15	T16	T17	T18	T19		T25	T26	T27	T28		
T13	T14	T15	T16	T17	T18	T23	T24	T25	T26	T27	T28	-----
T12	T13	T14	T15	T16	T17	T22	T23	T24	T25	T26	T27	

T11	T12	T13	T14	T15	T16	T21	T22	T23	T24	T25	T26	-----
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-------

FILTERING

FILTERING

T12	T11	T13	T15	T14	T21	T16	T22	T23	T24	T25	T26
-----	-----	-----	----------------	----------------	-----	----------------	-----	-----	-----	-----	-----

DISCARD

T12	T12	T13		T21	T21	T22	T23	T24	T25	T26
	T11	T12				T21	T22	T23	T24	T25
								T22	T23	T24

T11

T21	T21	T22	T23
		T21	T22
			T21

FIG. 9

