

ITALIAN PATENT OFFICE

Document No.

102012902095118A1

Publication Date

20140424

Applicant

STMICROELECTRONICS S.R.L.

Title

DISPOSITIVO E RELATIVO METODO PER LA SCRITTURA/LETTURA DI UN
REGISTRO DI MEMORIA CONDIVISO DA UNA PLURALITA DI PERIFERICHE.

TITOLO: “Dispositivo e relativo metodo per la scrittura/lettura di un registro di memoria condiviso da una pluralità di periferiche”.

DESCRIZIONE

Campo di applicazione

5 La presente descrizione ha per oggetto un dispositivo ed un metodo per la scrittura/lettura di un registro di memoria condiviso da parte di una pluralità di periferiche.

 In particolare la presente descrizione si riferisce ad un dispositivo digitale in grado di gestire la scrittura/lettura di un dato presente in un registro di memoria,
10 comune ad una pluralità di periferiche, allorquando due o più periferiche hanno la necessità di scrivere/leggere tale dato contemporaneamente.

 Il campo di applicazione della presente invenzione è, ad esempio, quello del settore automotive in cui vi è l'eventualità che due o più centraline elettroniche, presenti in un autoveicolo, abbiano la necessità di scrivere/leggere (ossia di accedere)
15 in contemporanea ad un dato memorizzato in una memoria che è condivisa tra le diverse centraline elettroniche.

Descrizione della tecnica anteriore

 E' noto nello stato della tecnica realizzare un sistema digitale avente un banco di memoria di tipo Ram che è ad esempio dotato di una doppia o multi porta
20 (dual/multi port) per accedere ai dati in esso memorizzati.

 L'operazione di scrittura/lettura dei dati memorizzati nel banco di memoria è garantita con un singolo segnale di clock, come ad esempio descritto nel documento EP 505779. Tuttavia, il sistema di tale documento EP 505779, ancorché permetta l'accesso, ad esempio a due periferiche al banco di memoria in accordo a stringenti
25 regole di temporizzazione, non è in grado di assicurare la possibilità di accedere nel

medesimo istante ai dati memorizzati nella memoria Ram.

Inoltre, il sistema descritto in tale documento EP 505779 non risulta essere scalabile, dato che l'aggiunta o la sottrazione della gestione di una periferica dal sistema digitale comporta la completa riprogettazione del sistema stesso.

5 In altre parole, il sistema del documento EP 505779 non consente di adeguarsi alle mutevoli esigenze di gestione del numero di periferiche una volta che il progetto del sistema digitale è stato deliberato.

SOMMARIO DELL'INVENZIONE

Uno scopo è quello di realizzare un dispositivo ed un metodo per la
10 scrittura/lettura di un registro di memoria da parte di una pluralità N di periferiche che sia in grado di soddisfare le esigenze prima riportate, in accordo con le caratteristiche rivendicate nella rivendicazione 1 e 8, rispettivamente.

Grazie ad una forma di realizzazione, è dunque possibile realizzare un dispositivo di lettura/scrittura che è in grado di gestire l'accesso ai dati memorizzati
15 in un registro di memoria da parte di diverse periferiche che operano secondo un proprio segnale di clock allorquando due o più periferiche hanno la necessità di accedere contemporaneamente.

Inoltre, grazie ad una forma di realizzazione, il dispositivo di lettura/scrittura è modulare dato che consente di gestire l'aggiunta o la sottrazione di periferiche senza
20 che questa operazione di aggiunta o sottrazione di periferiche comporti la ricostruzione dell'intero dispositivo. Infatti, aggiungere periferiche significa aggiungere ulteriori moduli di interfaccia uguali tra di loro al dispositivo di lettura/scrittura, dove le due tipologie di modulo di interfaccia sono composti da predefiniti elementi base.

25 Inoltre grazie al presente dispositivo, la priorità di accesso delle periferiche al

registro di memoria è mantenuta anche qualora vengano aggiunte/sottratte delle periferiche, dato che è sufficiente modificare l'assegnazione del grado di priorità a ciascun circuito di interfaccia, ossia cambiando l'ordine di connessione dei moduli di interfaccia. Questo perché qualora si voglia cambiare la priorità delle N periferiche, 5 basta semplicemente interfacciare tali N periferiche ad N data path (cfr. i moduli SL#1, ..., SL#N mostrati in figura 1) dell'unità centrale con un ordine diverso, secondo la regola per cui più è basso il numero di data path, più la priorità di periferica è alta.

La modularità del dispositivo permette di non modificare il codice RTL, anche 10 se la priorità di scrittura tra le periferiche N cambia in base alla funzionalità delle periferiche nel "System On Chip" (o brevemente SOC).

Inoltre grazie ad una forma di realizzazione il dispositivo di assemblaggio di ogni modulo è in grado di gestire la scrittura/lettura dei dati nel registro di memoria anche qualora le periferiche abbiano diverse priorità hardware di scrittura/lettura dei 15 dati memorizzati nel registro di memoria.

Inoltre secondo un aspetto il processo che permette la richiesta di espletare le operazioni di scrittura/lettura non presenta restrizioni di timing né restrizioni di software.

Infine il dispositivo di scrittura/lettura può essere realizzato a componenti 20 discreti ed implementato in accordo con la filosofia plug & play.

BREVE DESCRIZIONE DEI DISEGNI

Le caratteristiche ed i vantaggi della presente divulgazione risulteranno evidenti dalla seguente descrizione dettagliata di una possibile forma di realizzazione pratica, illustrata a titolo di esempio non limitativo nell'insieme dei disegni, in cui:

25 - la figura 1 mostra una vista schematica di principio della presente invenzione;

- la figura 2A mostra uno schema a blocchi secondo una possibile forma di realizzazione illustrata nella presente invenzione configurata per il caso in cui una sola periferica ha clock uguale ad un clock esterno $ck\#e$ al dispositivo per la scrittura/lettura di un registro di memoria condiviso da una pluralità di periferiche;

5 - la figura 2B mostra uno schema a blocchi dei due tipi di circuiti di modulo secondo una possibile forma di realizzazione, in un cui il primo comprende sette elementi base poiché il clock di periferica è diverso dal clock esterno ed il secondo tipo comprende cinque elementi base poiché il clock di periferica associato è uguale al clock esterno $ck\#e$;

10 - le figure da 3A/3B a 7A/7B mostrano, rispettivamente, uno schema a blocchi di una possibile forma di realizzazione di vari elementi di base del dispositivo ed un diagramma dei timing di funzionamento di tali elementi di base;

15 - la figura 8 mostra un diagramma di flusso del metodo di funzionamento del dispositivo per la scrittura/lettura di un registro di memoria condiviso da una pluralità di periferiche.

DESCRIZIONE DETTAGLIATA

Anche qualora non esplicitamente evidenziato, le singole caratteristiche descritte in riferimento alle specifiche realizzazioni dovranno intendersi come accessorie e/o intercambiabili con altre caratteristiche, descritte in riferimento ad altri
20 esempi di realizzazione.

La presente descrizione si riferisce ad un dispositivo di scrittura/lettura 1 di tipo digitale, implementabile in forma discreta o integrata in un cosiddetto “System On Chip” (o brevemente SOC), in cui vi sono N periferiche $Pe\#1, \dots, Pe\#N$, ciascuna delle quali opera, in generale con un proprio segnale di clock $ck\#1, \dots, ck\#N$,
25 asincrono rispetto ad un segnale di clock esterno $ck\#e$ e agli altri N-1 segnali di clock

di periferica.

In particolare, più periferiche possono avere un medesimo segnale di clock che a sua volta può essere diverso o uguale al clock esterno $ck\#e$.

Giova rilevare che nel prosieguo si è scelto trattare l'esempio raffigurato in
 5 figura 1 e 2A, in cui il segnale di clock $ck\#1$ è supposto essere uguale al segnale di clock esterno $ck\#e$, in modo da descrivere il funzionamento **di** due tipologie di modulo rappresentate nella figura 2B ed identificate con i riferimenti sl_7e e sl_5e , laddove:

- sl_7e rappresenta il modulo avente sette elementi base con il clock di
 10 periferica diverso dal clock esterno;

- sl_5e rappresenta il modulo avente cinque elementi base poiché il clock di periferica associato è uguale al clock esterno $ck\#e$, così da trascurare i due traslatori di clock del primo tipo di modulo.

Ciascuna delle N periferiche $Pe\#1, \dots, Pe\#N$ presenta un proprio valore di
 15 priorità di periferica Pr variabile tra un valore massimo $Pr_{max} = N$ ed un valore minimo $Pr_{min} = 1$.

Tali N periferiche $Pe\#1, \dots, Pe\#N$ devono potere scrivere/leggere (ossia condividere) un dato in un registro di memoria WM .

In particolare il dispositivo di scrittura/lettura 1 è configurato per gestire
 20 l'eventualità in cui due o più periferiche $Pe\#1, \dots, Pe\#N$, vogliano accedere nel medesimo istante al dato memorizzato in tale registro di memoria WM .

Ad esempio il numero N delle periferiche Pe può essere pari ad un numero intero arbitrario quale due, tre, quattro e così via.

Con riferimento alle figure annesse, il dispositivo per la scrittura/lettura 1
 25 comprende un'unità centrale 2 dotata del registro di memoria WM , tale unità centrale

2 risultando in comunicazione di segnale con un banco di moduli SL tramite i blocchi di interfaccia I/O.

In particolare, si ha che il banco di moduli SL, tramite i loro blocchi di interfaccia I/O, è in comunicazione di segnale con l'unità centrale 2 ed è configurato
5 per scrivere/leggere un dato $d\#i$ relativo a ciascuna della pluralità N di periferiche $Pe\#1, \dots, Pe\#N$ nel registro di memoria WM.

Secondo un aspetto, al fine di consentire di accedere al registro di memoria WM per scrivere/leggere il dato $d\#i$ da parte di due o più periferiche $Pe\#1, \dots, Pe\#N$ in contemporanea, il banco di moduli SL comprende una pluralità di moduli
10 $SL\#1, \dots, SL\#N$ che risultano essere predisposti in un numero pari alla pluralità N delle periferiche $Pe\#1, \dots, Pe\#N$.

In particolare, ciascuno di tali moduli $SL\#1, \dots, SL\#N$ tramite i loro blocchi di interfaccia di scrittura/lettura I/O risulta essere in comunicazione di segnale con una rispettiva periferica $Pe\#1, \dots, Pe\#N$.

15 Pertanto si ha che ciascun modulo $SL\#1, \dots, SL\#N$ è configurato per ricevere/scrivere/leggere il dato $d\#i$ generato dalla rispettiva periferica cui risulta essere connesso.

Ciascun modulo $SL\#1, \dots, SL\#N$ ha un valore di priorità Pr' che è variabile tra un valore massimo Pr_{max}' ed un valore minimo di priorità Pr_{min}' .

20 Giova rilevare, anche con riferimento alla figura 2A, che l'unità centrale 2 comprende un multiplexer MUX che è in comunicazione di segnale, da una parte con la pluralità di moduli $SL\#1, \dots, SL\#N$ e dall'altra con il registro di memoria WM.

In particolare la pluralità di moduli $SL\#1, \dots, SL\#N$ è connessa sia come driver al multiplexer MUX sia come receiver dal registro di memoria WM.

25 Secondo un aspetto della presente forma di realizzazione, la gestione delle

operazioni di accesso al dato $d\#i$ da parte delle N periferiche nel registro di memoria WM è eseguito in modo hardware attraverso la pluralità dei moduli $SL\#1, \dots, SL\#N$.

In particolare, ciascun modulo $SL\#1, \dots, SL\#N$ è in comunicazione di segnale con l'unità centrale 2 attraverso un proprio bus di collegamento di scrittura/lettura
 5 W/R Bus $\#1, \dots, W/R$ Bus $\#N$, con i variabile tra $1, \dots, N$.

Pertanto all'aumentare del numero N di periferiche da gestire da parte del dispositivo 1, questi deve prevedere altrettanti moduli $SL\#1, \dots, SL\#N$.

In particolare ciascun modulo della pluralità di moduli $SL\#1, \dots, SL\#N$ comprende una cella di arbitraggio AC, in modo tale che il primo modulo $SL\#1$ di
 10 detta pluralità di moduli $SL\#1, \dots, SL\#N$, sia identificato con il valore di priorità massimo Pr_{max}' e gli altri $N-1$ moduli $SL\#2, \dots, SL\#N$ sono identificati con valori di priorità decrescenti.

In altre parole si ha che al primo modulo $SL\#1$ è assegnato il valore di priorità massima Pr_{max}' , al secondo modulo $SL\#2$ è assegnato un valore di priorità inferiore
 15 rispetto al valore di priorità Pr_{max}' del primo modulo $SL\#1$, al terzo modulo $SL\#3$ è assegnato un valore di priorità inferiore rispetto al valore di priorità Pr_{max}' del primo modulo $SL\#1$ ed al valore di priorità del secondo modulo $SL\#2$, all' N -esimo modulo $SL\#N$ è assegnato un valore di priorità minimo Pr_{min}' che risulta essere inferiore rispetto al valore di priorità dell' $N-1$ -esimo modulo $SL\#N-1$.

20 La cella di arbitraggio AC, presente in ogni modulo, sia esso sl_7e o sl_5e , permette di escludere o abilitare l'accesso dello stesso al MUX in base al input ricevuto; pertanto ciascun modulo della pluralità di moduli $SL\#1, \dots, SL\#N$ ha una capacità di auto-arbitrazione della priorità.

Vantaggiosamente, secondo una forma preferita di realizzazione, la periferica
 25 $Pe\#h$ (laddove h è compreso tra $0 < h < N+1$), avente il valore di priorità massimo

Prmax, è quella destinata ad essere connessa al primo modulo SL#1 ossia del modulo avente il massimo di priorità Prmax'.

Le altre periferiche Pe sono connesse agli N-1 moduli di interfaccia secondo il loro valore di priorità.

5 In altre parole al modulo di interfaccia SL#2 (con l'indice "2" diverso dall'indice "h") è connessa la periferica Pe che ha la priorità più alta tra le rimanenti periferiche, dato che la periferica Pe#h avente il valore massimo di priorità è già stata connessa al modulo SL#1.

10 Secondo un aspetto l'unità centrale 2 opera ad una frequenza di clock principale CK che risulta essere predefinita e liberamente impostabile per scrivere/leggere il dato d#i presente nel registro di memoria WM, in quanto connesso esternamente al clock ck#e.

15 In particolare, a seconda di come si seleziona il clock esterno ck#e, il segnale di clock principale CK dell'unità centrale 2 può avere una frequenza pari alla frequenza del segnale clock di periferica ck#1,..., ck#N della periferica Pe connessa al modulo SL#1, ossia al modulo di interfaccia che opera con il valore di priorità massima Prmax. Quindi il segnale di clock principale CK dell'unità centrale 2 è pari al segnale di clock della periferica Pe che risulta essere connessa al modulo SL#1,..., SL#N cui è stato assegnato il valore di priorità massima Prmax.

20 Alternativamente l'unità centrale 2 può operare con un segnale di clock principale CK che ha una frequenza pari alla frequenza di un segnale di clock ck#e esterno al dispositivo, diversa da quella di tutti i clock ck#1,...,ck#N delle periferiche Pe. Ad esempio tale segnale di clock ck#e esterno può avere una frequenza superiore alla frequenza del segnale di clock ck#1,..., ck#N delle periferiche Pe.

25 Pertanto il segnale di clock principale CK, cui opera l'unità centrale 2 può

essere posto uguale alla frequenza, ossia connesso, ad uno qualunque degli $N+1$ clock disponibili, ossia gli N clock di periferica $ck\#1, \dots, ck\#N$ ed il clock esterno $ck\#e$.

Con riferimento ora alla Figura 8, secondo un aspetto del metodo è previsto di
 5 assegnare un valore di priorità massimo al primo modulo della pluralità di moduli $SL\#1, \dots, SL\#N$ ed un valore di priorità decrescente a ciascuno degli $N-1$ moduli della pluralità di moduli $SL\#1, \dots, SL\#N$ (blocco 7).

E' altresì previsto di connettere a ciascun modulo della pluralità di moduli $SL\#1, \dots, SL\#N$ una rispettiva periferica $Pe\#1, \dots, Pe\#N$ (blocco 8) e di imporre una
 10 frequenza di clock principale CK all'unità centrale 2. Preferibilmente tale frequenza di clock è posta pari alla frequenza di clock della periferica connessa al primo modulo $SL\#1$ della pluralità di moduli $SL\#1, \dots, SL\#N$ (blocco 9).

In particolare, secondo un aspetto del metodo, è previsto che il valore di priorità decresca sequenzialmente al crescere della numerazione degli $N-1$ moduli
 15 $SL\#2, \dots, SL\#N$.

Vantaggiosamente, dato che ciascuna periferica $Pe\#1, \dots, Pe\#N$ prevede un valore di priorità variabile tra un valore massimo ed un valore minimo, la fase di connettere ciascun modulo della pluralità di moduli $SL\#1, \dots, SL\#N$ ad una rispettiva periferica $Pe\#1, \dots, Pe\#N$ prevede di connettere al primo modulo a priorità massima
 20 la periferica avente priorità massima e agli altri $N-1$ moduli le rispettive periferiche a priorità decrescente.

Dato che il modulo $SL\#1$ ha la più alta priorità Pr di scrittura del registro di memoria WM, ossia $Pr_{max}=N$ e, dall'esempio di Figura 1 si ha che la periferica $Pe\#2$ risulta essere connessa al modulo $SL\#1$ tramite il blocco di scrittura e lettura
 25 I/O, allora nel modulo, la periferica $Pe\#2$ avrà la massima priorità di accesso rispetto

alle altre periferiche Pe.

Oltre a consentire l'accesso alle periferiche Pe#i con le loro priorità all'unico registro di memoria WM, è anche possibile gestire la situazione in accordo alla quale venga aggiunta un'ulteriore periferica Pe il cui valore, ad esempio di priorità, è superiore ai valori di priorità delle periferiche Pe sino ad ora gestite.

In particolare l'unità centrale 2 comprende una logica di controllo 6 e le celle di arbitraggio AC risultano essere disposte secondo uno schema di tipo daisy chain, ossia come una catena modulare di celle di arbitraggio.

Tali celle arbitraggio AC risultano essere in comunicazione di segnale con la logica di controllo 6 per valutare, in funzione del valore di priorità Pr' dei moduli SL#1, ..., SL#N, quale dato d#i di ciascuna delle pluralità N di periferiche Pe#1, ..., Pe#N sia da scrivere nel registro di memoria WM.

Con riferimento ora alle Figura 2A e 2B si nota che la pluralità di moduli SL#1, ..., SL#N è composta da circuiti di interfaccia uguali tra loro.

Giova rilevare che ciascun modulo si concretizza nell'impiegare i medesimi componenti circuitali; ossia le N periferiche Pe#1, ..., Pe#i si interfacciano con i rispettivi moduli SL#1, ..., SL#N che risultano essere tra di loro identici nella loro forma di realizzazione circuitale.

Ciascun modulo SL#1,..., SL#N qualora avesse il clock connesso esternamente al pin CK#J diverso dal clock connesso esternamente al pin CK ossia $ck\#e \neq ck\#i$, allora è di tipo sl_7e e comprende sette elementi circuitali (come da Figura 2B); mentre qualora il clock di modulo connesso esternamente al pin CK#J è uguale al clock connesso esternamente al pin CK ossia con $ck\#e = ck\#i$, allora ciascuno degli N moduli è di tipo sl_5e e comprende cinque elementi circuitali (come da Figura 2B).

Con riferimento ancora alla Figura 2A, il fatto che il modulo SL#1 sia stato scelto di tipo sl_5e, ossia composto di soli cinque elementi base (vedi Figura 2B), comporta che il segnale di clock esterno ck#e sia uguale al segnale di clock ck#2, ossia della periferica Pe#2.

5 Il modulo SL#2, che nell'esempio di Figura 1 è connesso alla periferica Pe#N, ha una priorità inferiore a quella modulo SL#1, ma superiore alla priorità degli altri moduli, ed opera alla frequenza del segnale di clock ck#N di periferica Pe#N tramite la connessione al pin CK#2 del modulo SL#2.

10 Il fatto che il clock ck#N abbia una frequenza diversa da ck#e, significa che correttamente il modulo SL#2 è stata scelta di tipo sl_7e, ossia composta da sette "elementi base" (come mostrato in Figura 2B).

In vista di ciò e con riferimento alla Figure 2A e 2B, ciascuno modulo SL#1, ..., SL#N comprende gli elementi di base che costituiscono il modulo di tipo sl_7e e/o il modulo sl_5e.

15 In particolare il modulo sl_5e comprende almeno:

- una interfaccia Write Input W_I che è configurata per consentire a ciascuna di detta pluralità N di periferiche di fornire il rispettivo segnale di clock di periferica ck#1,..., ck#N e il rispettivo dato d#i a ciascuno modulo di interfaccia ed
- un registro Write Slave WSR configurato per memorizzare il dato d#i della
20 rispettiva periferica, prima che tale dato d#i venga caricati nel registro di memoria WM;
- una cella di arbitraggio AC (Arbitration Cell) è configurata in daisy chain per valutare quando il dato d#i disponibile nel Write Slave Register WSR deve essere registrato nel registro di memoria WM.
- 25 - un Read Register RR configurato per visualizzare i contenuto del registro di

memorizzazione WR ed

- un'interfaccia Read Output R_O configurata per consentire una comunicazione di segnale alla rispettiva periferica Pe così da leggere il contenuto del registro di memoria WM.

5 In particolare il modulo sl_7e comprende, oltre a quanto previsto nel modulo In sl_5e, almeno:

- una cella Write Request/Acknowledge Process WRAP configurata per trasferire il dato d#i dal segnale di clock di periferica ck#i al segnale di clock principale CK.

10 - una cella Read Request/Acknowledge Process RRAP configurata per trasferire i dati presenti nel registro di memoria WM che lavora al segnale di clock di periferica ck#1,..., ck#N al registro Read Register RR che opera al segnale di clock principale CK.

L'unità centrale 2 comprende una logica di controllo 6 in modo tale che il
15 multiplexer MUX riceva in ingresso sequenzialmente i dati d#i dallo Write Slave Register WSR in funzione sia della cella di arbitraggio AC che della temporizzazione della logica di controllo 6, quest'ultima operando con il segnale di clock principale CK e ponga in uscita il dato d#i che deve essere scritto nel registro di memoria WM.

Giova rilevare che il segnale di clock principale CK è propagato all'intero del
20 perimetro del blocco 2A ossia tra la cella Write Request/Acknowledge Process WRAP e la cella Read Request/Acknowledge Process RRAP di ciascun modulo SL#j, mentre all'esterno di tale perimetro è previsto che si operi alla frequenza di clock di periferica (ossia esternamente al blocco 2A di Figura 2A).

Con riferimento alla specifica forma di implementazione della Figura 2A, il
25 clock principale su CK è uguale al clock della periferica Pe#2 connessa al modulo

SL#1 di tipo sl_5e, per cui il clock su CK risulta di fatto uguale al clock su CK#1.

In tale fattispecie non è necessario traslare i dati d#i che operano con il clock della periferica Pe#1 al registro di memoria WM, così che il circuito del modulo SL#1 di tipo sl_5e risulta essere privo della cella Write Request/Acknowledge
 5 Process WRAP e della cella Read Request/Acknowledge Process RRAP. Questi due blocchi elementari sono invece presenti nei rimanenti moduli SL#j (j=2..N) tutti di tipo sl_7e (Figure 2A e 2B).

Qualora più periferiche debbano scrivere contemporaneamente i rispettivi dati d#i nel registro di memoria WM e, dato che alla prima pluralità di circuiti
 10 d'interfaccia di scrittura viene assegnato un valore di priorità variabile tra un massimo ed un minimo, ad esempio il valore massimo è assegnato al modulo SL#1, e agli altri moduli SL#2, ..., SL#j ..., SL#N è assegnato un valore man mano decrescente, il valore di priorità più elevato è assegnato alla periferica Pe#i che è connessa al modulo SL#j che presenta il più basso valore di j, ossia a quel modulo
 15 SL#j che presenta il maggiore valore di priorità escludendo il modulo SL#1, per cui il successivo è j=2.

Qualora sia necessario un clock principale CK più veloce rispetto a quello delle singole periferiche Pe1, ..., Pe#i, ..., Pe#N, è possibile ad esempio connettere il clock principale CK ad un dedicato clock esterno ck#e che risulti essere più veloce
 20 rispetto ai clock ck#1,..., ck#i, ..., ck#N delle periferiche, mantenendo la priorità della periferica Pe#i interfacciata a SL#1 al valore più alto.

A tale fine, sempre con riferimento alla Figura 2A, si nota che la periferica Pe#i carica il suo dato d#i, nel modulo SL#j attraverso il bus W Bus#j. Nel medesimo tempo la periferica Pe#i trasferisce il dato d#i quando il segnale XWait#j sale ad un
 25 valore alto, come mostrato nello schema a blocchi di implementazione

dell'interfaccia Write Input W_I di figura 4A.

Il dato D#j (il dato circolante nello slot SL#j), ossia il dato d#i trasmesso dalla periferica Pe#i e circolante all'interno del modulo SL#j, è mantenuto nell'interfaccia Write Input W_I fino a quando l'operazione di scrittura di tutti i registri R_O è confermata attraverso la cella WRAP del rispettivo modulo SL#j. Allora la periferica Pe#ij trasferisce il dato d#i quando il segnale XW#j va basso.

Secondo un aspetto, il modulo SL#j opera con un segnale di clock di periferica ck#i connesso al pin CK#j, mentre il registro WM opera con il segnale di clock principale CK. Tramite un processo di richiesta/autorizzazione, il circuito WRAP del rispettivo modulo SL#j carica il dato D#j temporizzati al segnale di clock di periferica ck#i connesso al pin CK#j, all'interno del registro di memoria WM che opera con il segnale di clock principale.

Giova rilevare che il registro WSR del modulo SL#1 nell'esempio riportato in figura 2A, non necessita della cella WRAP e RRAP, dato che è stato ipotizzato che il segnale di clock principale CK è forzato esternamente il modulo dallo stesso segnale di clock di periferica, (ck#2 nell'esempio di figura 1) connesso al modulo SL#1 tramite il pin CK#1 di figura 2B (ck#e=ck#2).

Qualora il segnale di clock principale CK sia connesso ad un segnale di clock esterno ck#e diverso da ogni altro clock di periferica, allora tutti i moduli SL#j prevedono di implementare la cella WRAP e RRAP perché dovranno muovere il dato D#j da un circuito sincronizzato da CK#j (=ck#i) ad un circuito sincronizzato da CK (=ck#e) e viceversa.

Quando il registro di memoria WM è aggiornato con i dati della periferica Pe#i, allora una qualunque cella RRAP#j carica il dato D#j all'interno del proprio registro RR#j.

Attraverso un processo di richiesta/autorizzazione, la cella $RRAP\#j$ consente sia di leggere il contenuto del registro di memoria WM alla frequenza del segnale di clock principale CK che di scrivere i dati $D\#j$ in un registro che opera al segnale di clock di periferica $ck\#i$.

- 5 Infine il contenuto $D\#j$ del registro di memoria WM è scritto in tutte le celle R_O così che tutte le periferiche $Pe\#i$ possono leggerlo.

Giova rilevare, come descritto in seguito, che durante la fase di lettura di una qualunque delle periferiche $Pe\#i$ il segnale $XR\#j$ sale ad un valore alto.

- Nel prosieguo sono descritti possibili forme di implementazione dei
10 componenti e loro forme d'onda di funzionamento facenti parte dei moduli $SL\#1, \dots, SL\#N$ del dispositivo 1, con l'accortezza che i segnali non specificamente descritti sono da ritenere come provenienti dalla logica di controllo 6.

- Con riferimento ora alle figure 3A e 3B, in cui sono mostrate rispettivamente una possibile forma di realizzazione della cella W_I di uno dei moduli $SL\#j$ e dei
15 suoi timing di funzionamento, si nota che tale cella $W_I\#j$ trasferisce il dato $D\#j$ scritto dalla rispettiva periferica $Pe\#i$ nel registro di memoria WM in accordo con il proprio bus di trasferimento.

Secondo un aspetto sia la periferica $Pe\#i$ che la cella W_I del circuito del modulo $SL\#j$ operano al medesimo segnale di clock $ck\#i = CK\#j$.

- 20 Il registro $WI_Buffer\#j$ riceve un insieme di dati validi $D\#j$, attraverso il bus di periferica $WBUS\#j$, quando la periferica $Pe\#i$ disabilita il segnale di controllo di scrittura $Bus_Write\#j$, garantendo un valore stabile sul bus $Bus_Write_Data\#j$.

- Se durante la fase di scrittura, il registro $WI_Buffer\#j$ è diverso rispetto al contenuto registro di memoria WM, visualizzata attraverso il registro Read Output
25 $RO_Reg\#j$ della cella R_O del modulo $SL\#j$, si ha che:

1) il contenuto del buffer $WI_Buffer\#j$ sarà trasferito all'uscita della cella W_I del modulo $SL\#j$, ossia al registro Write Input $WI_Reg\#j$;

2) la cella WRAP del modulo $SL\#j$, riceve l'impulso di start da WI_Write per iniziare il trasferimento dei dati del $WI_Reg\#j$ nel registro di memoria WM
5 attraverso una richiesta/conferma del processo;

3) la salita del segnale WI_XWait interrompe qualunque accesso di lettura/scrittura della periferica $Pe\#i$ al modulo $SL\#j$.

Una volta che i dati $WI_Reg\#j$ sono stati scritti nel registro di memoria WM e nei registri Slave Read (vedi $Read_Slave_CK\#j$ in figura 6A), il segnale
10 $Fall_WI_XWait\#j$ asserisce e la periferica $Pe\#i$ può accedere al modulo $SL\#j$.

Con riferimento ora alle figure 4A e 4B, in cui sono mostrate rispettivamente una possibile forma di realizzazione della cella WRAP di uno dei moduli $SL\#j$ e dei suoi timing di funzionamento, si nota che la cella WRAP del modulo $SL\#j$ trasferisce i dati $d\#i$ della periferica $Pe\#i$ presenti sul registro $WI_Reg\#j$ che opera col clock
15 $ck\#j$, nel registro WSR del modulo $SL\#j$, cadenzato dal segnale di clock principale CK. Il cambiamento del dominio di clock avviene attraverso una richiesta/conferma del protocollo handshake che permette di trasferire il dato $D\#j$ da un dominio di clock asincrono $CK\#j$ ad un altro dominio di clock asincrono CK.

Una volta che il registro $WI_Reg\#j$ riceve un nuovo dato $D\#j$, il segnale
20 WI_XWait (vedere Figura 4A) sale ad un valore alto, un impulso viene effettuato sul segnale $Load_WSR_j_CK\#j$ (vedi figura 4A e 4B) ed il registro $Write_Slave\#j$ viene aggiornato con i nuovi dati $D\#j$.

Il registro $Write_Slave\#j$, che opera sotto il dominio del segnale $CK\#j$, mantiene i nuovi dati $D\#j$ stabili, fintanto che il processo di richiesta li ha copiati nel
25 registro $Write_Slave\#JCK$ sotto il dominio del clock principale CK.

Il segnale Load_WSR#j_CK produce l'impulso di trasferimento durante un sicuro ciclo di clock principale CK, dove sia il registro Write_Slave#j ed il registro Write_Slave#JCK non sono guidati da altri impulsi. Poi, fino a quando il registro di memoria WM è stato scritto con il valore del registro Write_Slave#JCK, il registro

5 Write_Slave#JCK mantiene il dato D#j stabile.

Una volta che il registro Read_SlaveCK#j (vedi in figura 6A) è aggiornato dal contenuto del registro di memoria WM, il segnale Fall_WI_XWait#j è valido per un ciclo di clock principale CK pulendo il registro Write_Slave#jCK.

Con riferimento ora alle figure 5A e 5B, in cui sono mostrate rispettivamente

10 una possibile forma di realizzazione della cella di arbitraggio AC di uno dei moduli SL#j e dei suoi timing di funzionamento, si nota che tale cella di arbitraggio AC trasferisce il dato D#j proveniente dalla periferica Pe#i nel registro di memoria WM.

Il dominio del segnale di clock è quello del clock principale CK.

Se nello stesso ciclo di clock, più registri Write_Slave richiedono di scrivere il

15 registro di memoria WM, la periferica Pe con il valore più basso scriverà per prima il registro di memoria WM.

Il dato sarà diffuso a tutti i registri R_O da una richiesta/conferma di processo di tipo handshake e nel frattempo il processo di arbitraggio è bloccato.

La selezione della massima priorità è elaborata attraverso una catena modulare

20 di celle di arbitraggio AC.

In particolare, secondo un aspetto preferito, la cella di arbitraggio del modulo SL#j AC compara il suo valore di priorità con la precedente cella SL#(j-1). AC e informa la prossima cella SL#(j+1). AC se l'arbitraggio è vinto oppure no.

Giova rilevare che, tenendo in considerazione che il registro Slave Write

25 registra l'arbitraggio quando un solo dato D#j è pronto per essere scritto nel registro

di memoria WM (vedi figura 5A per la vista schematica e la figura 5B per le forma d'onda, dove si ipotizza vincitore $\#j=1$), quando l'impulso Load_WSR $\#j$ CK è valido, il nuovo dato D $\#j$ è scritto sul bus Write_Slave#JCK e la cella di arbitraggio AC Arbri_Cell $\#j$ del modulo SL $\#j$ ha in ingresso il segnale Set_Chan $\#j$.

- 5 Il canale $\#j$ vince l'arbitraggio e l'uscita Xt_Chain $\#j$ della cella di arbitraggio AC del modulo SL $\#j$ è disabilitata.

Questo significa anche, che nessuna altra cella di arbitraggio AC può vincere lungo la catena di arbitraggio, perché i rispettivi valori sono forzati ad un valore basso dal segnale Win_Chain $\#j$.

- 10 Nel ciclo successivo viene memorizzato il dato D $\#j$ nel registro di memoria WM ed il bit di attesa Pend_Chan $\#j$ viene cancellato; tutte le richieste incominciano ad aggiornare i registri R_O, laddove il segnale R_A sale ad un valore alto disabilitando tutta la catena di arbitraggio.

- Quando le celle R_O vengono aggiornate dal contenuto del registro di memoria WM, il segnale R_A scende verso un valore basso dato che le procedure per la richiesta sono terminate.

- Qualora due o più dati sono pronti per essere scritti, al medesimo ciclo di clock principale CK, nel registro di memoria WM, anche con riferimento alla figura 5B, il segnale Load_WSR $\#1$ _CK può essere considerato come il canale di alta priorità e il segnale Load_WSR $\#j$ _CK può essere considerato come il canale di priorità inferiore. Entrambi possono impostare il bit in sospeso, ma la catena di arbitraggio viene disabilitata per il canale $\#j$ fino a che tutte le celle R_O non sono aggiornate con il contenuto del registro di memoria WM.

- Dopo che il segnale R_A scende verso un valore basso, l'arbitraggio sarà vinto dal canale più a monte con il bit in attesa ancora impostato.

Successivamente, il comportamento sarà lo stesso descritto precedentemente in relazione al caso in cui un solo dato è pronto per essere scritto nel registro di memoria WM in un ciclo di clock principale CK.

Con riferimento ora alle figure 6A e 6B, in cui sono mostrate rispettivamente
 5 una possibile forma di realizzazione della cella RRAP di uno dei moduli SL#j sia per il caso in cui $j=1$ che per il caso in cui $j>1$ e dei suoi timing di funzionamento, si nota che la cella RRAP trasferisce i dati presenti nel registro di memoria WM, operando alla frequenza del segnale di clock principale CK, al registro Read_SlaveCK#j, che opera al segnale di clock di periferica ck#i.

10 Il trasferimento dei dati tra due diversi domini di clock asincroni avviene attraverso una richiesta/conferma del protocollo handshake.

In particolare nel caso in cui $j=1$ si ha che il registro Read_Master viene aggiornato con un ciclo di clock principale CK dopo che il registro di memoria WM è stato aggiornato (vedi la parte superiore della figura 6A) perché $CK\#j=CK$.

15 La periferica Pe#2, connessa al modulo SL#1 in base all'esempio portato in figura 1, è in grado di leggere una volta che i dati saranno caricati nella cella di interfaccia R_O attraverso il bus RI_Reg#1bus.

In particolare nel caso i moduli di interfaccia presentano un segnale di clock uguale a quello del clock principale CK ossia nel caso in cui $j>1$ si ha che per
 20 aggiornare tutti gli altri registri SL#j.Read_SlaveCK#J (vedi la parte inferiore in figura 6A), l'impulso Load_RS#j_CK#j è valido attraverso una procedura di richiesta/riconoscimento, laddove il contenuto del registro Read_Master (operante alla frequenza del segnale di clock CK) è copiato in sicurezza nel registro Read_SlaveCK#j (operante alla frequenza del segnale di clock ck#i).

25 La periferica Pe#j collegata al modulo di interfaccia SL#j, sarà in grado di

leggere i dati, quando i dati sono stati caricati nella cella R_O attraverso il bus RI_Reg#j bus.

Quando tutte le celle RRAP dei moduli SL#1, ..., SL#j hanno aggiornato i loro registri Read_SlaveCK#1,...,N, allora tutti i segnali Write Waits sono disabilitati, vale
 5 a dire tutti i segnali SL#1.WI_Xwait#j,..., SL#N.WI_Xwait#j scendono ad un valore basso attraverso il controllo della logica di controllo 6.

Questo permette al registro di memoria WM di essere scritto.

Con riferimento ora alle figure 7A e 7B, in cui sono mostrate rispettivamente una possibile forma di realizzazione della cella R_O di uno dei moduli SL#j e dei
 10 suoi timing di funzionamento, si nota che la cella R_O permette di copiare il contenuto del registro di memoria WM nel registro RO_Reg#j, da cui la periferica Pe#i coinvolta può leggere attraverso il bus di uscita Bus_Read_Data#j, quando il segnale di controllo Bus_Read#j è valido.

I dati presenti nel registro di memoria WM non sono direttamente leggibile
 15 dalla periferica Pe#i attraverso il bus RI_Reg#j, perché se fosse così potrebbe cambiare lo stato della periferica Pe#i, andando a porre in tale bus un valore imprevedibile.

Per questo motivo, ad ogni successivo ciclo di clock di periferica ck#i, l'uscita del registro RO_Reg#j viene aggiornata, come mostrato nelle forme d'onda di figura
 20 7B.

La periferica Pe#i, al fine di leggere il registro di memoria WM attraverso il bus Bus_Read_Data#j, deve rendere valido il segnale di controllo Bus_Read#j che, una volta attivato forza un refresh del registro RO_Reg#j.

In tal modo è garantita una lettura sicura del registro di memoria WM da parte
 25 della periferica Pe#i.

Poiché un tempo pari al ciclo di clock di periferica $ck\#i$ è necessario per aggiornare il registro $RO_Reg\#j$, i segnali di controllo $RO_XWait\#j$ vengono portati ad un valore alto per un ciclo di clock di periferica $ck\#i$ su qualsiasi periferica $Pe\#i$.

In particolare, nella figura 7B, sono riportati le forme d'onda relative a due successive richieste di lettura da parte delle periferiche $Pe\#i$.

Giova rilevare che sia i segnali $RO_XWait\#j$ che i segnali $WI_XWait\#j$ sono posti in OR rispetto al segnale $Bus_XWait\#j$ che appartiene al bus di interfaccia della periferica $Pe\#i$.

Giova rilevare che secondo un aspetto, al fine di garantire un corretto scrittura/lettura ad ogni periferica $Pe\#i$, eventuali inappropriati accessi sono gestiti attraverso il segnale di Transfer Wait allorquando è posto ad un valore alto, così da interrompere le operazioni di nuove scritture del registro di memoria WM fino a quando l'attuale operazione di scrittura non è completata.

Ovviamente, un tecnico del ramo, allo scopo di soddisfare esigenze contingenti e specifiche, potrà apportare numerose modifiche e varianti allo strumento ottico ed al sistema di visualizzazione prima descritti, tutte peraltro contenute nell'ambito di protezione quale definito dalle seguenti rivendicazioni.

RIVENDICAZIONI

1. Dispositivo digitale per la scrittura/lettura di un dato ($d\#i$) presente in un registro di memoria (WM), condiviso da parte di una pluralità N di periferiche ($Pe\#1, \dots, Pe\#N$) ciascuna periferica avendo un segnale di clock di periferica ($ck\#1, \dots, ck\#N$,
 5 allorquando due o più di detta pluralità di periferiche ($Pe\#1, \dots, Pe\#N$) hanno la necessità di scrivere/leggere tale dato ($d\#i$) contemporaneamente, detto dispositivo digitale comprendente un'unità centrale (2) avente detto registro di memoria (WM) ed un banco di moduli (SL) in comunicazione di segnale con detta unità centrale (2), detto banco di moduli (SL), essendo configurato per scrivere/leggere detto dato ($d\#i$),
 10 **caratterizzato dal fatto che** detto banco di moduli (SL) comprende una pluralità di moduli ($SL\#1, \dots, SL\#N$) aventi un valore di priorità (Pr') variabile tra un valore massimo (Pr_{max}') ed un valore minimo di priorità (Pr_{min}') e ciascuno di detti moduli ($SL\#1, \dots, SL\#N$) essendo connesso ad una rispettiva periferica ($Pe\#1, \dots, Pe\#N$), detta unità centrale (2) comprende un multiplexer (MUX) in comunicazione
 15 di segnale da una parte con detta pluralità di moduli ($SL\#1, \dots, SL\#N$) e dall'altra con detto registro di memoria (WM), ciascun modulo di detta pluralità di moduli ($SL\#1, \dots, SL\#N$) comprendendo una cella di arbitraggio (AC) configurata in modo tale che il primo modulo ($SL\#1$) di detta pluralità di moduli ($SL\#1, \dots, SL\#N$) sia identificato con detto valore di priorità massimo (Pr_{max}') e gli altri N-1 moduli
 20 ($SL\#2, \dots, SL\#N$) di detta pluralità di moduli ($SL\#1, \dots, SL\#N$) siano identificati con valori di priorità decrescenti, detta unità centrale (2) operando ad una predeterminata frequenza di clock principale (CK) per scrivere/leggere detto dato ($d\#i$) presente in detto registro di memoria (WM).
2. Dispositivo digitale per la scrittura/lettura in accordo con la rivendicazione 1, in
 25 cui detta unità centrale comprende una logica di controllo (6) e dette celle di

arbitraggio (AC) sono disposte come una catena modulare di celle di arbitraggio e risultano essere in comunicazione di segnale con detta logica di controllo (6) per valutare, in funzione di detto valore di priorità (Pr') di ciascuno di detti moduli (SL#1, ..., SL#N), quale dato (d#i) di ciascuna di detta pluralità N di periferiche (Pe#1, ..., Pe#N) è da scrivere/leggere in detto registro di memoria (WM).

3. Dispositivo digitale per la scrittura/lettura in accordo con la rivendicazione 1 o 2, in cui detta predeterminata frequenza di clock principale (CK) è pari ad una frequenza di un segnale di clock esterno a detto dispositivo digitale, detti moduli (SL#1, ..., SL#N) operando a detta frequenza di clock principale (CK).

4. Dispositivo digitale per la scrittura/lettura in accordo con la rivendicazione 3, ciascun modulo (SL#1, ..., SL#N) comprende almeno:

- un'interfaccia Write Input (W_I) che è configurata per consentire a ciascuna di detta pluralità (N) di periferiche di fornire il rispettivo segnale di clock di periferica (ck#1,..., ck#N) e il rispettivo dato (d#i) a ciascuno modulo di interfaccia;

- un registro Write Slave (WSR) configurato per memorizzare il dato (d#i) della rispettiva periferica, prima che tale dato (d#i) venga caricato nel registro di memoria WM;

- una cella di arbitraggio (AC) è configurata in daisy chain per valutare quando il dato (d#i) disponibile nel Write Slave Register (WSR) deve essere registrato nel registro di memoria (WM);

- un Read Register (RR) configurato per visualizzare il contenuto del registro di memoria (WM) ed

- un'interfaccia Read Output (R_O) configurata per consentire una comunicazione di segnale alla rispettiva periferica (Pe) così da leggere il contenuto del registro di memoria (WM).

5. Dispositivo digitale per la scrittura/lettura in accordo con la rivendicazione 4, in cui ciascun modulo (SL#1, ..., SL#N) comprende:

- una cella Write Request/Acknowledge Process (WRAP) configurata per trasferire il dato (d#i) dal segnale di clock di periferica (ck#i) al segnale di clock principale (CK).

- una cella Read Request/Acknowledge Process (RRAP) configurata per trasferire i dati presenti nel registro di memoria (WM) che lavora al segnale di clock di periferica (ck#1,..., ck#N) al registro Read Register (RR) che opera al segnale di clock principale (CK).

6. Dispositivo digitale per la scrittura/lettura in accordo con la rivendicazione 4, in cui detto multiplexer (MUX) è configurato per ricevere in ingresso sequenzialmente i dati di detto Write Slave Register (WSR) in funzione sia di cella di arbitraggio (AC) che della temporizzazione di detta logica di controllo (6), quest'ultima operando con detto segnale di clock principale (CK).

7. Dispositivo digitale per la scrittura/lettura in accordo con la rivendicazione 6, in cui detta cella di arbitraggio (AC) trasferisce il dato (d#i), che proviene dalla relativa periferica (Pe#1, ..., Pe#N) in detto registro di memoria (WM), detta cella di arbitraggio (AC) lavorando con detto clock principale (CK), detti dato (d#i) essendo trasmessi a detta interfaccia Read Output (R_O) in funzione di un arbitraggio di tipo handshake.

8. Metodo per la scrittura/lettura di un registro di memoria (WM) condiviso da una pluralità di periferiche (Pe#1, ..., Pe#N) ciascuna periferica avendo un segnale di clock di periferica (ck#1,..., ck#N), detto metodo comprendendo le fasi di:

- disporre di un'unità centrale (2) avente un registro di memoria (WM) ed un banco di moduli (SL) in comunicazione di segnale con detta unità centrale (2), detto banco

di moduli (SL) essendo configurato per scrivere/leggere un dato (d#i) relativo a ciascuna di detta pluralità N di periferiche (Pe#1, ..., Pe#N) in detto registro di memoria (WM), detto metodo **è caratterizzato dal fatto di:**

- disporre una pluralità di moduli (SL#1, ..., SL#N) in un numero pari alla pluralità
- 5 N di dette periferiche (Pe#1, ..., Pe#N);
- numerare sequenzialmente da uno ad N ciascun modulo pluralità di moduli (SL#1, ..., SL#N);
- assegnare un valore di priorità massima al primo modulo di detta pluralità di moduli (SL#1, ..., SL#N);
- 10 - assegnare un valore di priorità decrescente a ciascuno degli N-1 moduli di detta pluralità di moduli (SL#1, ..., SL#N);
- connettere a ciascun modulo di detta pluralità di moduli (SL#1, ..., SL#N) una rispettiva periferica (Pe#1, ..., Pe#N);
- imporre una frequenza di clock principale (CK) a detta unità centrale (2).
- 15 **9.** Metodo per la scrittura/lettura di un registro di memoria (WM) condiviso da una pluralità di periferiche (Pe#1, ..., Pe#N) in accordo con la rivendicazione 8, in cui detta fase di assegnare un valore di priorità decrescente prevede che il valore di priorità decresca sequenzialmente al crescere della numerazione degli N-1 moduli (SL#1, ..., SL#N).
- 20 **10.** Metodo per la scrittura/lettura di un registro di memoria (WM) condiviso da una pluralità di periferiche (Pe#1, ..., Pe#N) in accordo con la rivendicazione 8, in cui ciascuna periferica (Pe#1, ..., Pe#N) prevede un valore di priorità variabile tra un valore massimo ed un valore minimo, detta fase di connettere a ciascun modulo di detta pluralità di moduli (SL#1, ..., SL#N) una rispettiva periferica (Pe#1, ..., Pe#N)
- 25 prevede di connettere a detto primo modulo a priorità massima la periferica avente

priorità massima e agli altri N-1 moduli di interfaccia rispettive periferiche a priorità decrescente.

11. Metodo per la scrittura/lettura di un registro di memoria (WM) condiviso da una pluralità di periferiche (Pe#1, ..., Pe#N) in accordo con la rivendicazione 8, in cui
 5 una cella di arbitraggio (AC#j) compara il valore di priorità del rispettivo modulo di interfaccia con il valore di priorità della precedente cella (AC#(j-1)) e informa la prossima cella di arbitraggio AC#(j +1) se l'arbitraggio è vinto oppure no.

12. Metodo per la scrittura/lettura di un registro di memoria (WM) condiviso da una pluralità di periferiche (Pe#1, ..., Pe#N) in accordo con la rivendicazione 8, in cui
 10 detta frequenza di clock principale (CK) è posta pari alla frequenza di clock di detta periferica connessa a detto primo modulo (SL#1) di detta pluralità di moduli (SL#1, ..., SL#N).

CLAIMS

1. A digital device for writing/reading a piece of data (d#i) in a memory register (WM) shared by a plurality N of peripherals (Pe#1, ..., Pe#N), each peripheral having a peripheral clock signal (ck#1,..., ck#N), when two or more of said plurality
5 of peripherals (Pe#1, ..., Pe#N) need to write/read such piece of data (d#i) at the same time, said digital device comprising a central unit (2) having said memory register (WM) and a bank of modules (SL) in signal communication with said central unit (2), said bank of modules (SL) being designed to write/read said piece of data (d#i), **characterized in that** said bank of modules (SL) comprises a plurality of
10 modules (SL#1,..., SL#N) whose priority value (Pr') ranges between maximum (Prmax') and minimum (Prmin') priority values, each of said modules (SL#1, ..., SL#N) being connected to a respective peripheral (Pe#1, ..., Pe#N), said central unit (2) comprises a multiplexer (MUX) in signal communication on the one hand with said plurality of modules (SL#1,..., SL#N), and on the other hand with said memory
15 register (WM), each module of said plurality of modules (SL#1, ..., SL#N) comprising an arbitration cell (AC) which is designed such that the first module (SL#1) of said plurality of modules (SL#1, ..., SL#N) is identified by said maximum priority value (Prmax') and the other N-1 modules (SL#2,...,SL#N) of said plurality of modules (SL#1, ..., SL#N) are identified by decreasing priority values, said
20 central unit (2) operating at a predetermined main clock frequency (CK) to write/read said piece of data (d#i) in said memory register (WM).
2. A digital writing/reading device as claimed in claim 1, wherein said central unit comprises a control logic (6) and said arbitration cells (AC) are arranged in a modular chain of arbitration cells and are in signal communication with said control
25 logic (6) to ascertain, according to said priority value (Pr') of each of said modules

(SL#1, ..., SL#N), which piece of data (d#i) of each of said plurality N of peripherals (Pe#1, ..., Pe#N) is to be written/read into/from said memory register (WM).

3. A digital writing/reading device as claimed in claim 1 or 2, wherein said predetermined main clock frequency (CK) is equal to a frequency of a clock signal
5 external to said digital device, said modules (SL#1, ..., SL#N) operating at said main clock frequency (CK).

4. A digital writing/reading device as claimed in claim 3, each module (SL#1, ..., SL#N) comprising at least:

- a Write Input interface (W_I), which is designed to allow each of said
10 plurality (N) of peripherals to provide its respective peripheral clock signal (ck#1,..., ck#N) and its respective piece of data (d#i) to each interface module;

- a Write Slave Register (WSR), which is designed to store the piece of data (d#i) of the corresponding peripheral, before loading of such piece of data (d#i) into the memory register (WM);

15 - an arbitration cell (AC) is configured in daisy chain arrangement, to ascertain when the piece of data (d#i) available in the Write Slave Register (WSR) has to be stored into the memory register (WM);

- a Read Register (RR), which is designed to display the contents of the memory register (WM) and

20 - a Read Output interface (R_O), which is designed to allow signal communication with the corresponding peripheral (Pe), to read the contents of the memory register (WM).

5. A digital writing/reading device as claimed in claim 4, wherein each module (SL#1, ..., SL#N) comprises:

25 - a Write Request/Acknowledge Process cell (WRAP), which is designed to

transfer the piece of data (d#i) from the peripheral clock signal (ck#i) to the main clock signal (CK);

- a Read Request/Acknowledge Process (RRAP), which is designed to transfer the data in the memory register (WM) that operates at the peripheral clock signal (ck#1,..., ck#N) , to the Read Register (RR) that operates at the main clock signal (CK).

6. A digital writing/reading device as claimed in claim 4, wherein said multiplexer (MUX) is designed to sequentially receive at its input the data of said Write Slave Register (WSR), according to both the arbitration cell (AC) and the timing of said control logic (6), the latter operating with said main clock signal (CK).

7. A digital writing/reading device as claimed in claim 6, wherein said arbitration cell (AC) transfers the piece of data (d#i) that comes from the corresponding peripheral (Pe#1, ..., Pe#N) to said memory register (WM), said arbitration cell (AC) operating with said main clock (CK), said data (d#i) being transmitted to said Read Output interface (R_O) according to an arbitration of handshake type.

8. A method of writing/reading a memory register (WM) shared by a plurality of peripherals (Pe#1, ..., Pe#N) each peripheral having a peripheral clock signal (ck#1,..., ck#N), said method comprising the steps of:

- providing a central unit (2) having a memory register (WM) and a bank of modules (SL) in signal communication with said central unit (2), said bank of modules (SL) being designed to write/read a piece of data (d#i) for each of said plurality (N) of peripherals (Pe#1, ..., Pe#N) in said memory register (WM), said method is **characterized in that:**

- providing a plurality of modules (SL#1, ..., SL#N) which are as many as the plurality N of said peripherals (Pe#1, ..., Pe#N);

- sequentially numbering each module of the plurality of modules (SL#1, ..., SL#N) from one to N;
 - assigning a maximum priority value to the first module of said plurality of modules (SL#1, ..., SL#N);
 - 5 - assigning decreasing priority values to each of the N-1 modules of said plurality of modules (SL#1, ..., SL#N);
 - connecting each module of said plurality of modules (SL#1, ..., SL#N) with a respective peripheral (Pe#1, ..., Pe#N);
 - setting a main clock frequency (CK) for said central unit (2).
- 10 **9.** A method of writing/reading a memory register (WM), shared by a plurality of peripherals (Pe#1, ..., Pe#N) as claimed in claim 8, wherein said step of assigning decreasing priority values requires the priority value to sequentially decrease as the numbers of the N-1 modules (SL#1, ..., SL#N) increase.
- 10.** A method of writing/reading a memory register (WM) shared by a plurality of
- 15 peripherals (Pe#1, ..., Pe#N) as claimed in claim 8, wherein each peripheral (Pe#1, ..., Pe#N) has a priority value ranging between a maximum value and a minimum value, said step of connecting each module of said plurality of modules (SL#1,..., SL#N) to a respective peripheral (Pe#1, ..., Pe#N) includes connecting said first maximum-priority module with the maximum-priority peripheral and the other N-1
- 20 modules with respective peripherals having decreasing priorities.
- 11.** A method of writing/reading a memory register (WM), shared by a plurality of peripherals (Pe#1, ..., Pe#N) as claimed in claim 8, wherein an arbitration cell (AC#j) compares the priority value of its respective interface module with the priority value of the previous cell (AC#(j-1)) and informs the next arbitration cell
- 25 AC#(j +1) whether arbitration was won or not.

12. A method of writing/reading a memory register (WM), shared by a plurality of peripherals (Pe#1, ..., Pe#N) as claimed in claim 8, wherein said main clock frequency (CK) is set to be equal to the clock frequency of said peripheral connected to said first module (SL#1) of said plurality of modules (SL#1, ..., SL#N).

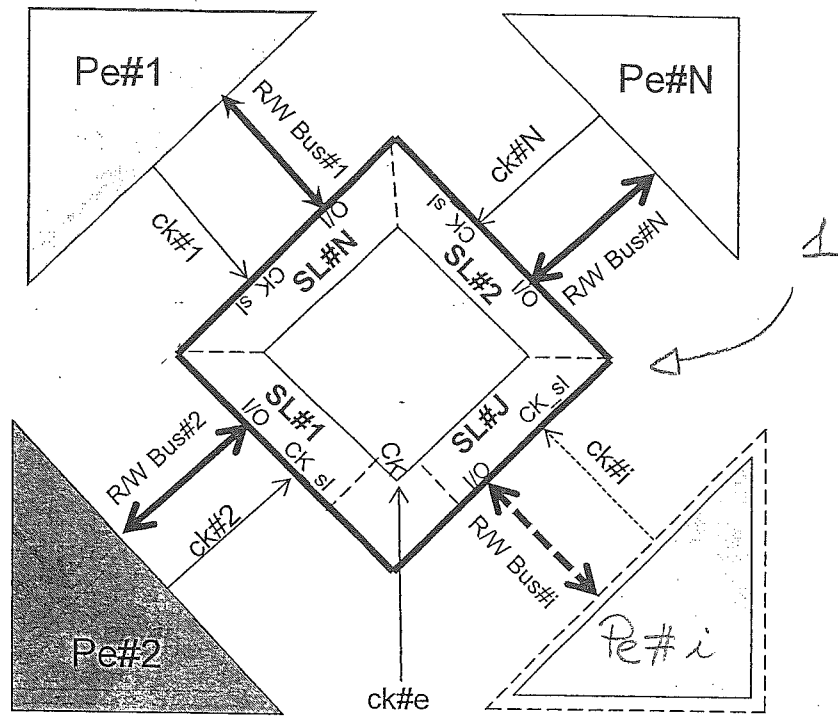


FIG. 1

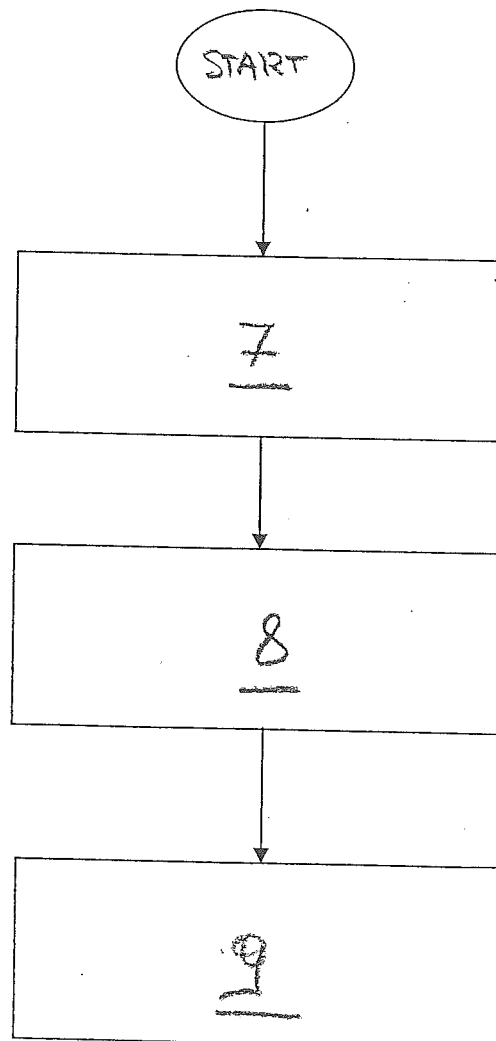


FIG. 8

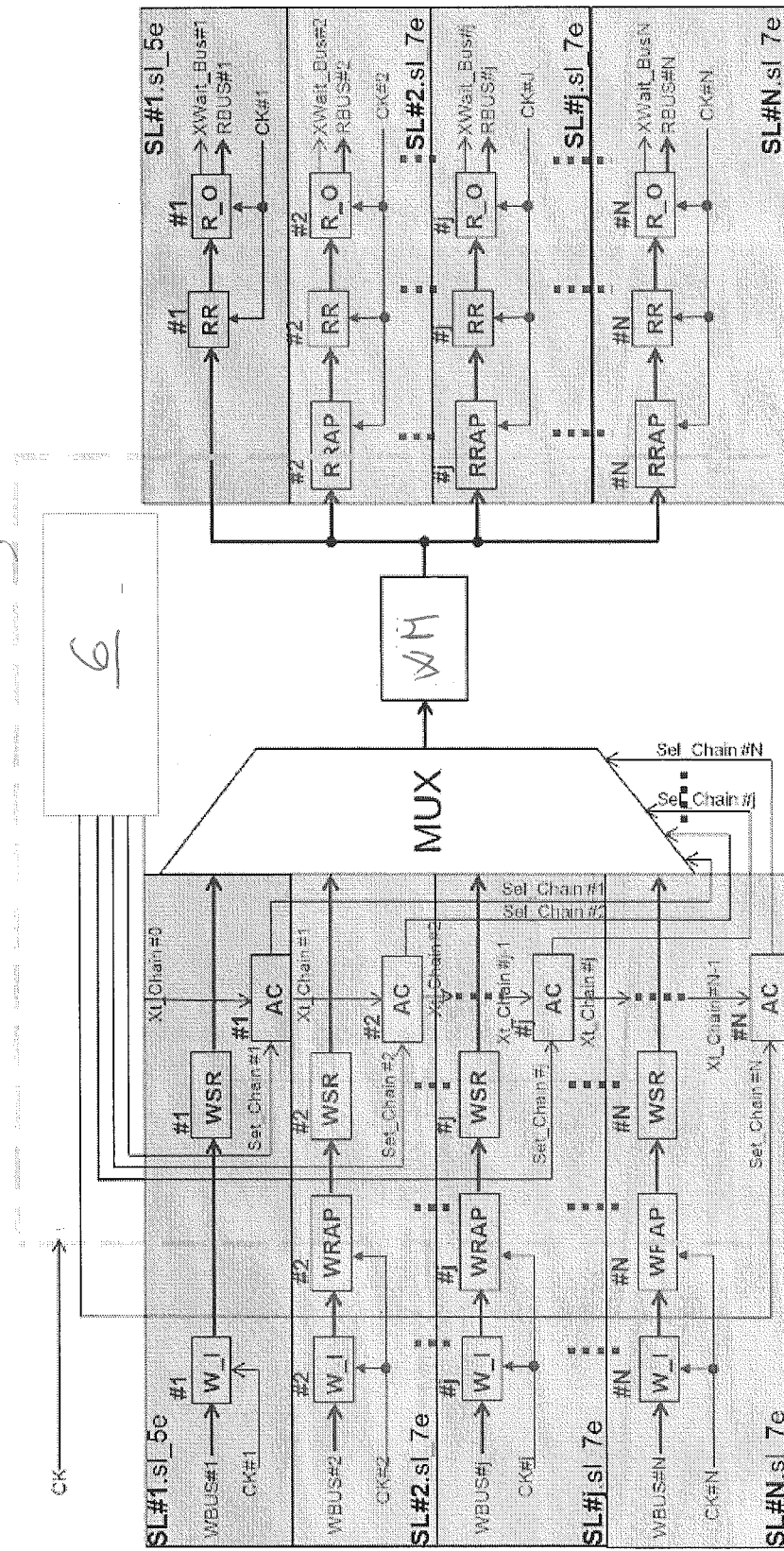


Fig. 2A

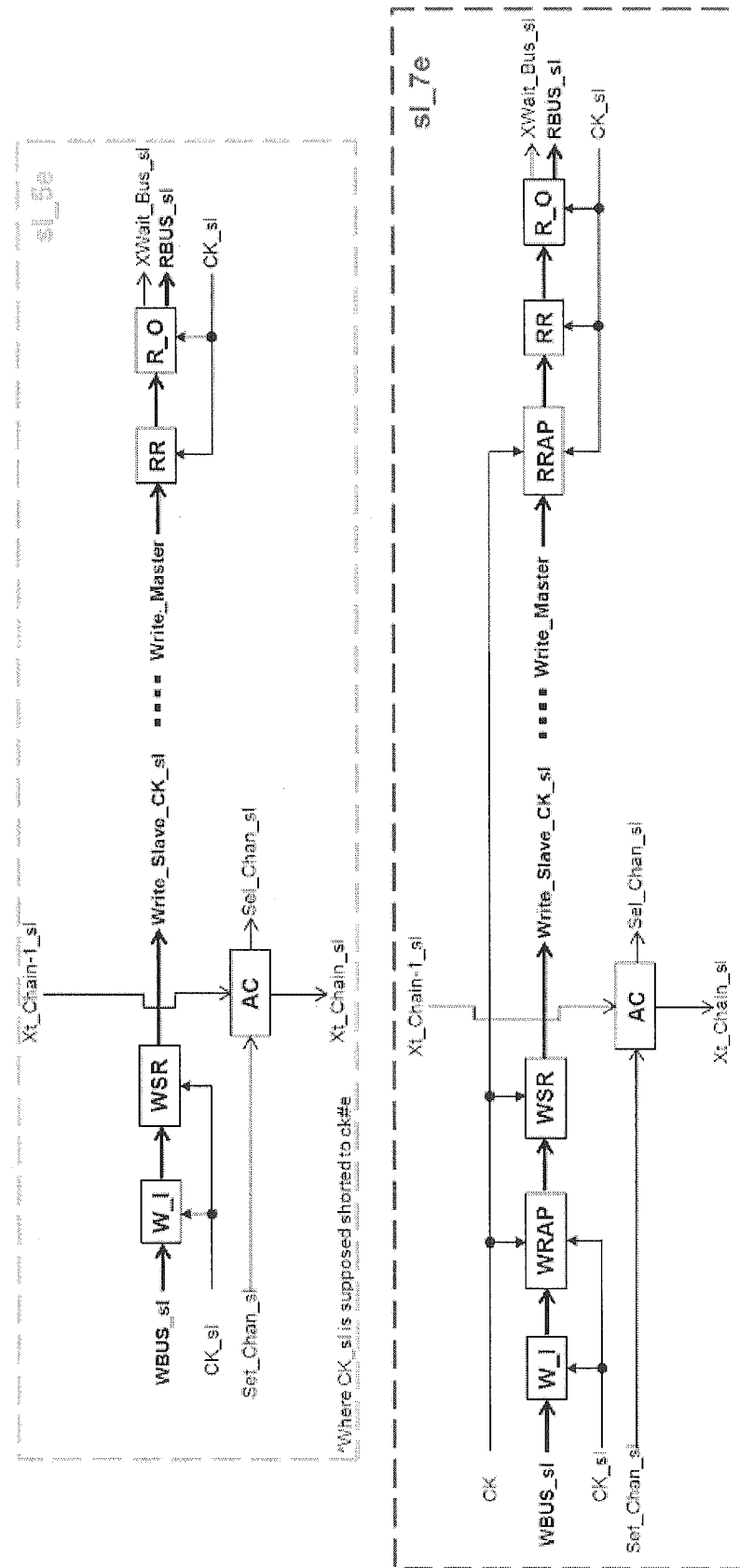


Fig. 2B

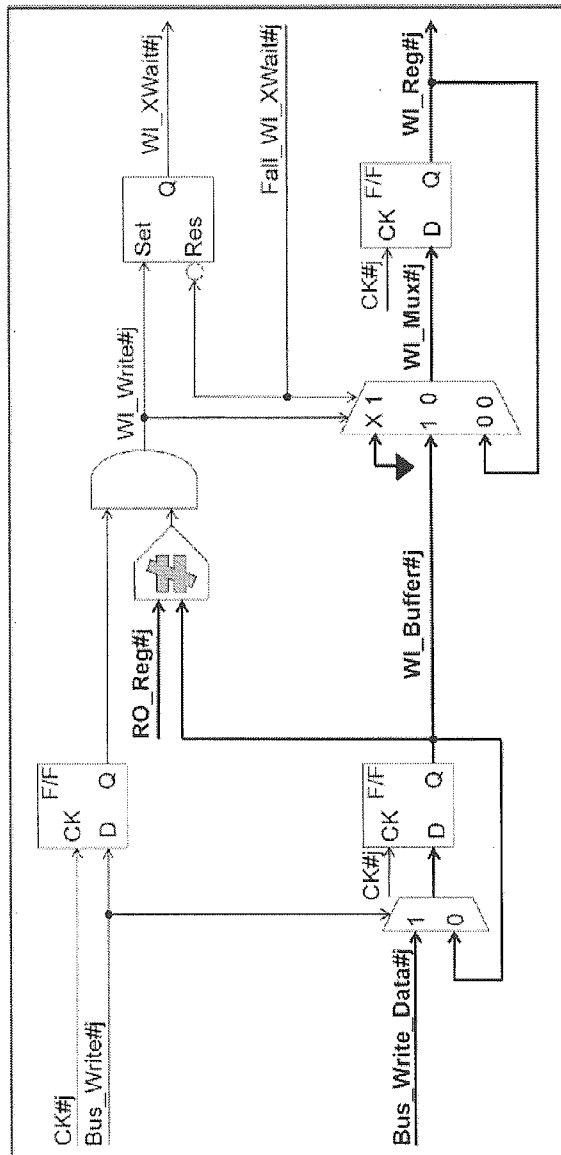


Fig. 3A

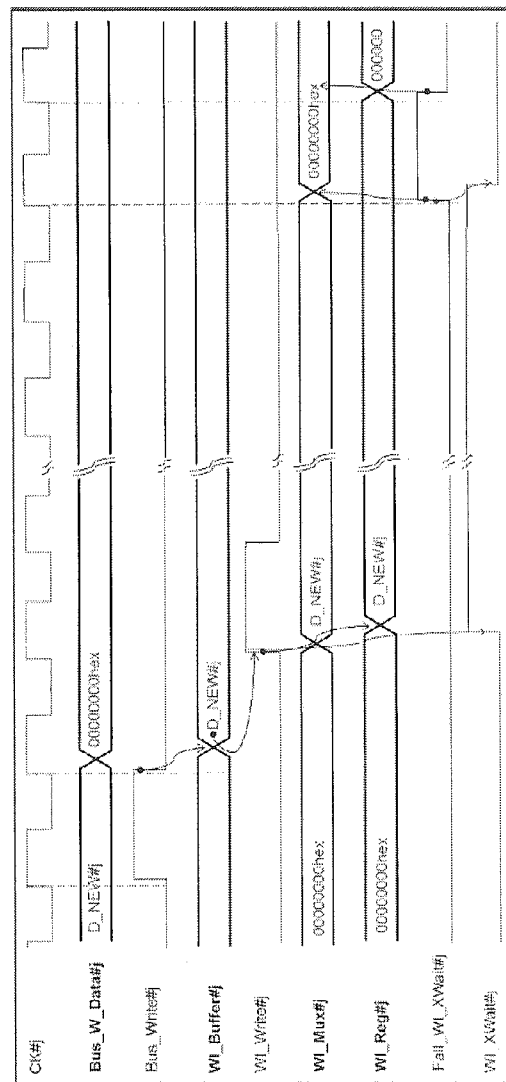


Fig. 3B

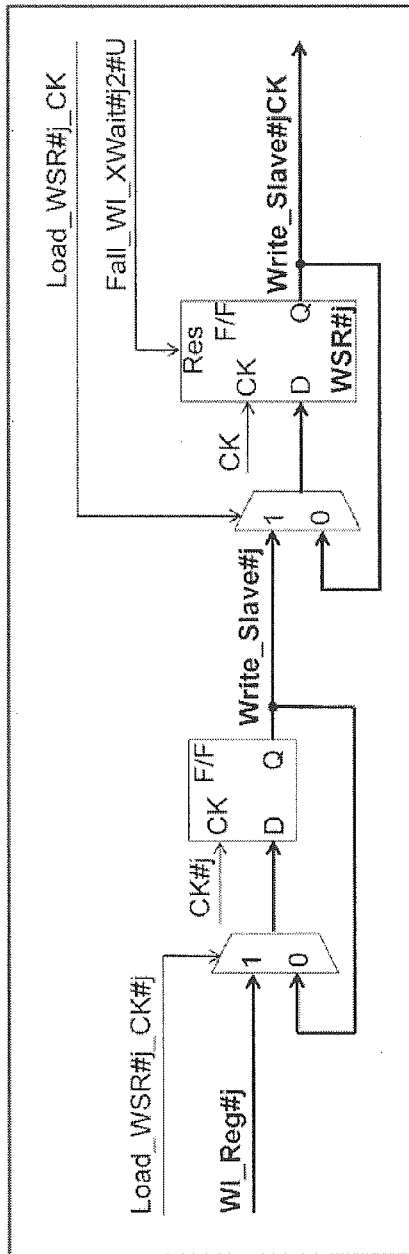


Fig. 4A

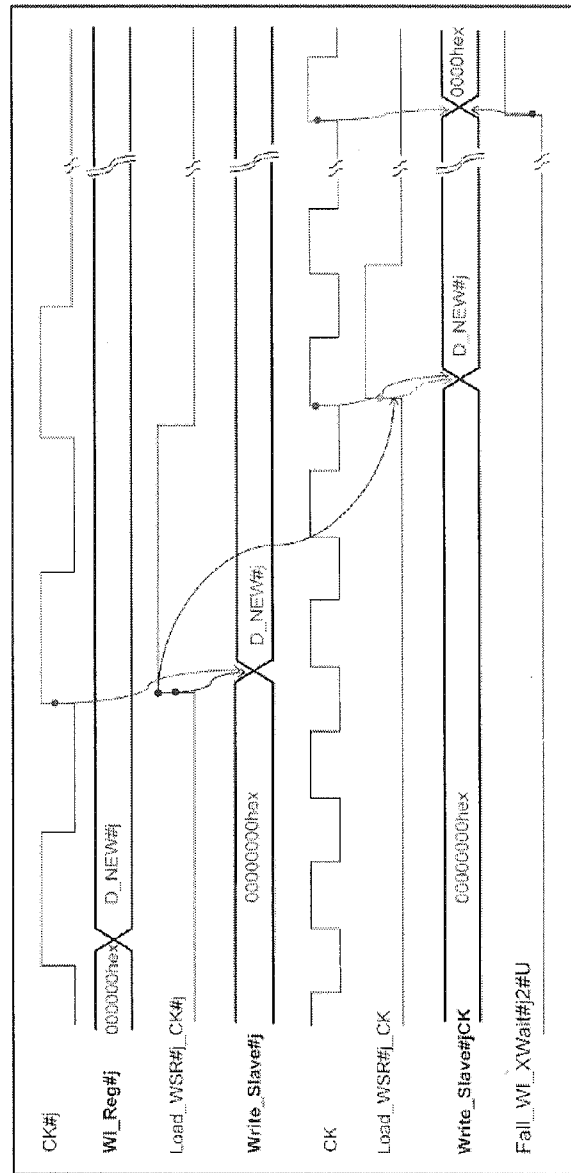


Fig. 4B



Fig. 5A

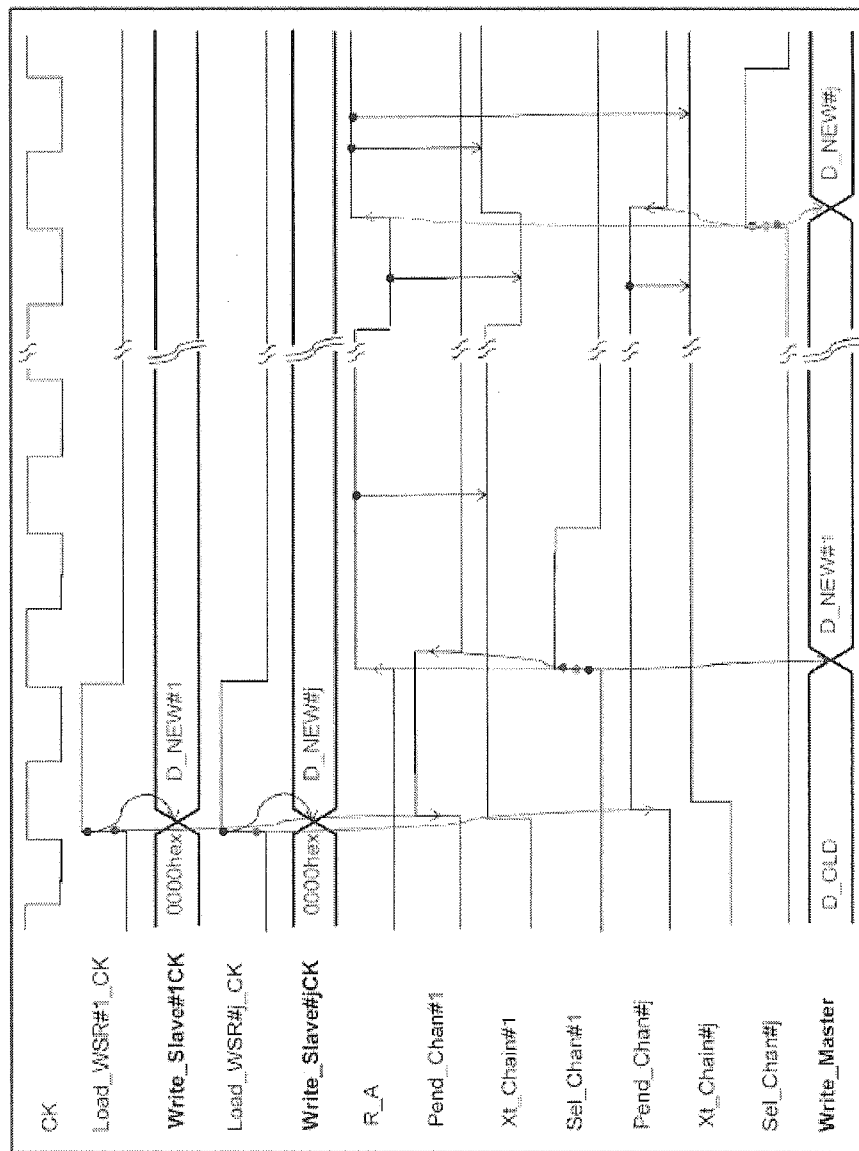


Fig. 5B

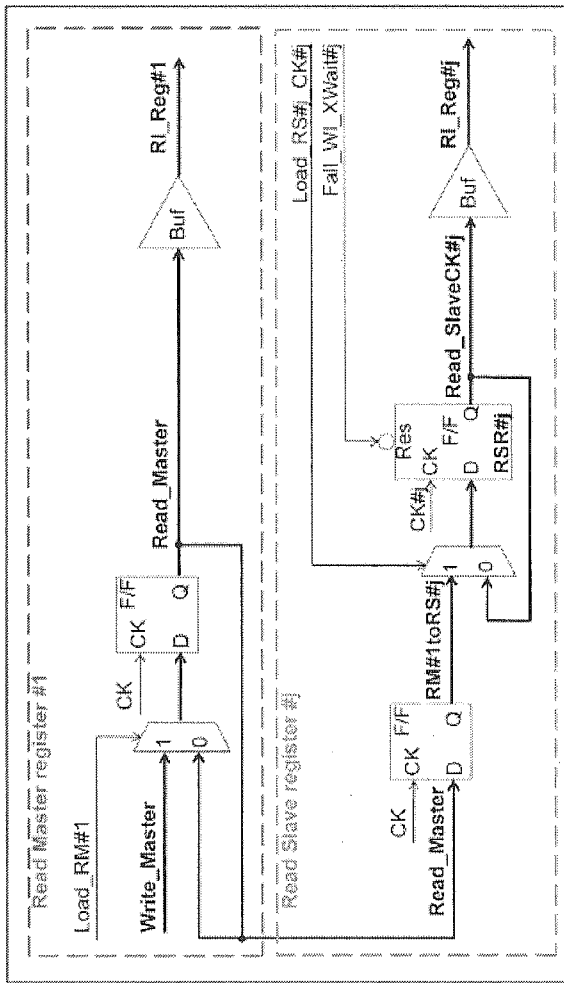


Fig. 6A

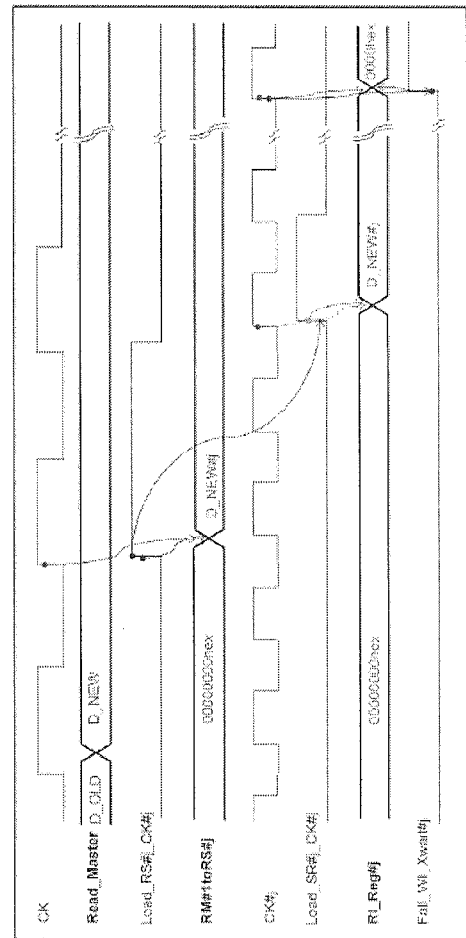


Fig. 6B

