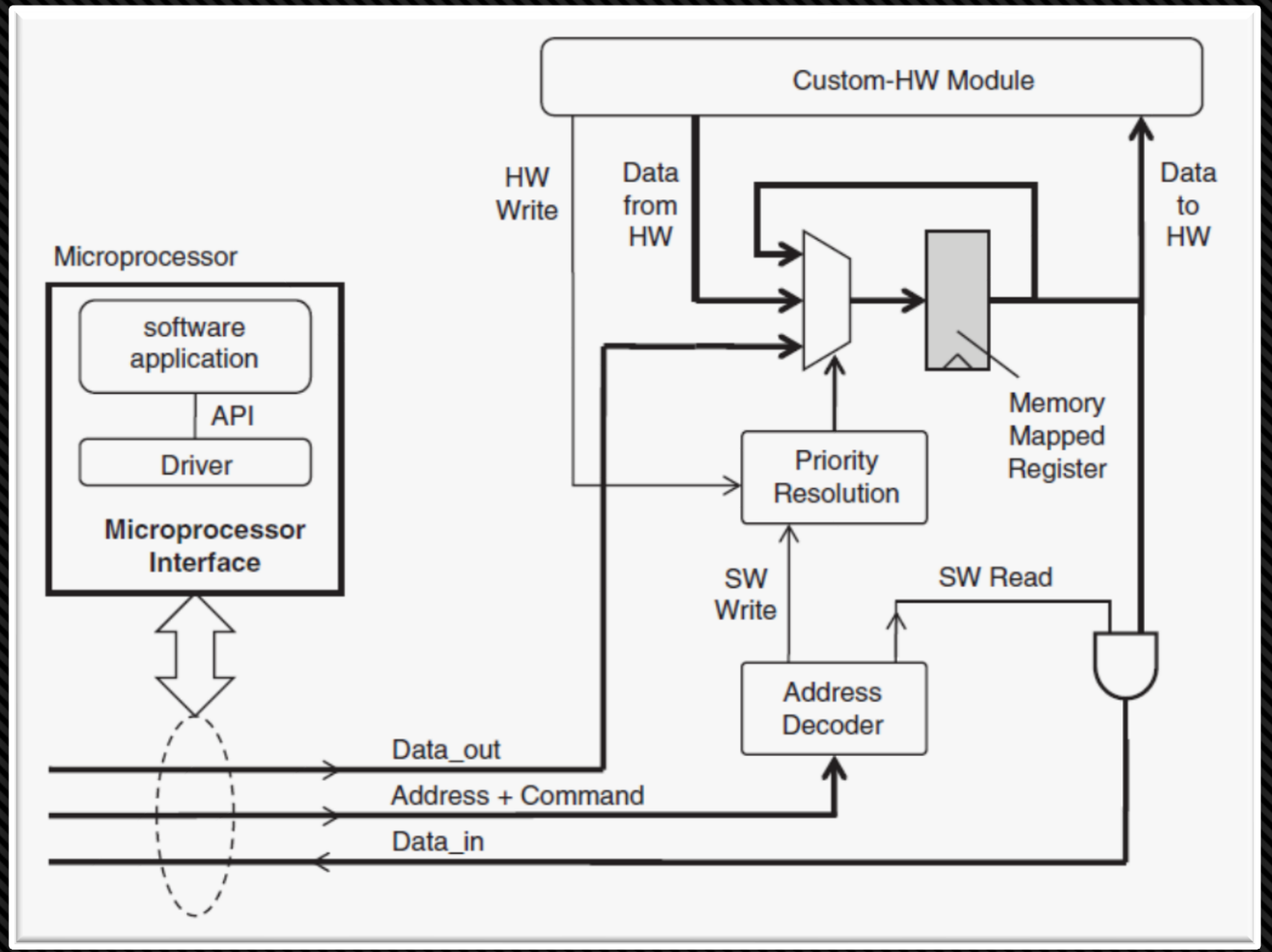
INTERFACCE DI MICROPROCESSORE

INTERFACCE MAPPATE IN MEMORIA

- LE INTERFACCE HARDWARE/SOFTWARE MAPPATE IN MEMORIA SONO LE Più GENERICHE E DIFFUSE. DAL PUNTO DI VISTA SOFTWARE ESSE VENGONO GESTITE MEDIANTE L'UTILIZZO DI PUNTATORI.

INTERFACCIA MAPPATA IN MEMORIA PER REGISTRI

- Viene usato il bus di trasferimento che si trova sul chip
- Sarà necessario un decoder degli indirizzi
- Servirà un decoder per assegnare le priorità ed evitare situazioni di conflitto
- Si devono attuare delle strategie per far sì che i dati vengano salvati solo nel registro in memoria principale ed evitare che si trovino in cache o nei registri del processore.


Custom-HW Module
HW
Write
Data
from
HW
Data
to
HW
Microprocessor
software
application
API
Driver
Microprocessor
Interface
Memory
Mapped
Register
Priority
Resolution
SW
Write
SW Read
Address
Decoder
Data_out
Address + Command
Data_in

11.1 Memory-Mapped Interfaces 321

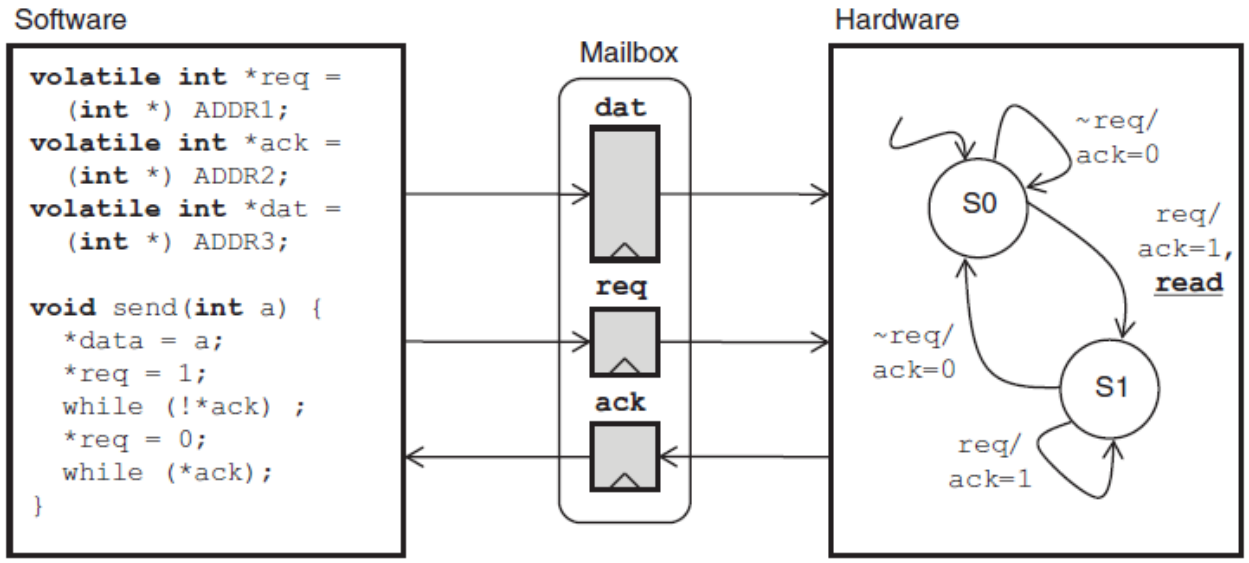


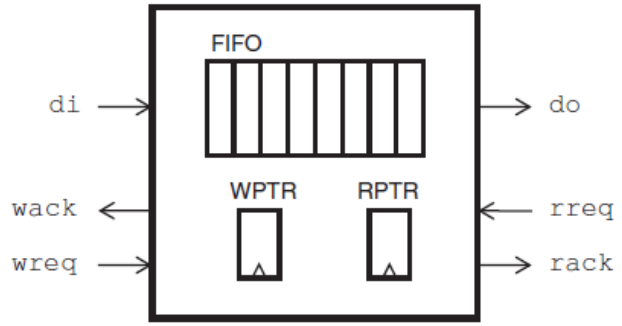
Fig. 11.3 A mailbox register between hardware and software

Mailboxes

- Le mailboxes sono un'estensione dell'interfacce mappate in memoria per registri
- Prevedono una fase di handshake tra HW e SW con l'utilizzo di due flag di req e ack
- Il protocollo ha due punti di sincronizzazione: quando entrambi i flag sono settati a 0 e quando sono up.
- Per via dei continui handshakes l'overhead su performance e design è elevato.
- Si adatta al modulo più lento.
- Interazione che segue un flusso ben preciso.

FIFO

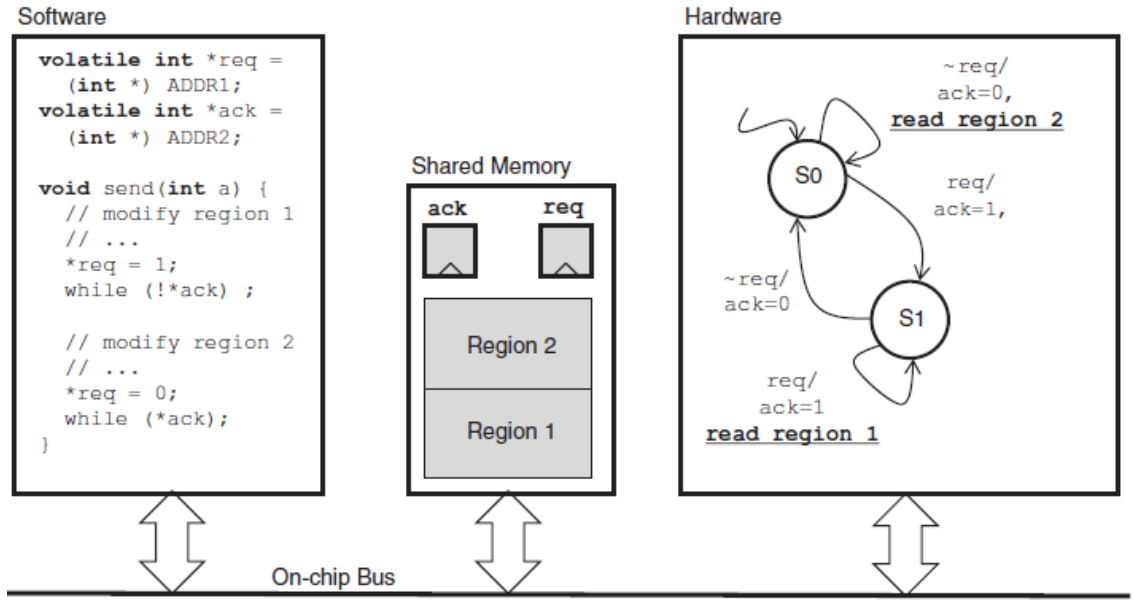
- In realtà le operazioni di lettura e scrittura hanno nature molto differenti tra loro ecco perché conviene separarle progettando una struttura che preveda l'utilizzo di due coppie di segnali (req,ack), una per operazioni.
- La coda(FIFO) permette di immagazzinare i tokens di scrittura e smistarli regolarmente in lettura.
- La coda è caratterizzata da tre stati: empty, notempty e full.
- La fifo presenta le interfacce read e write con ruolo slave.
- Si può pensare di impostare l'interfaccia come master.




```
dp fifo(in  di    : ns(8);
        in  wreq  : ns(1);
        out wack  : ns(1);
        out do    : ns(8);
        in  rreq  : ns(1);
        out rack  : ns(1)) {
  sig read, write : ns(1);
  reg rptr, wptr : ns(3);
  use dualport_mem(di, wptr, write,  // write port
                   do, rptr, read);  // read port
  always {
    read  = (rreq & (wptr != rptr)) ? 1 : 0;
    write = (wreq & ((wptr + 1) != rptr) ? 1 : 0;
    wptr  = write ? wptr + 1 : wptr;
    rptr  = read ? rptr + 1 : rptr;
    wack  = write;
    rack  = read;
  }
```

Shared memory

- Piuttosto di usare un meccanismo di handshake per un singolo registro si può pensare di implementarlo per un range dello spazio di indirizzamento.
- La regione di memoria sarà partizionata in due, ogni parte interverrà in fasi diverse del protocollo così da garantire consistenza.

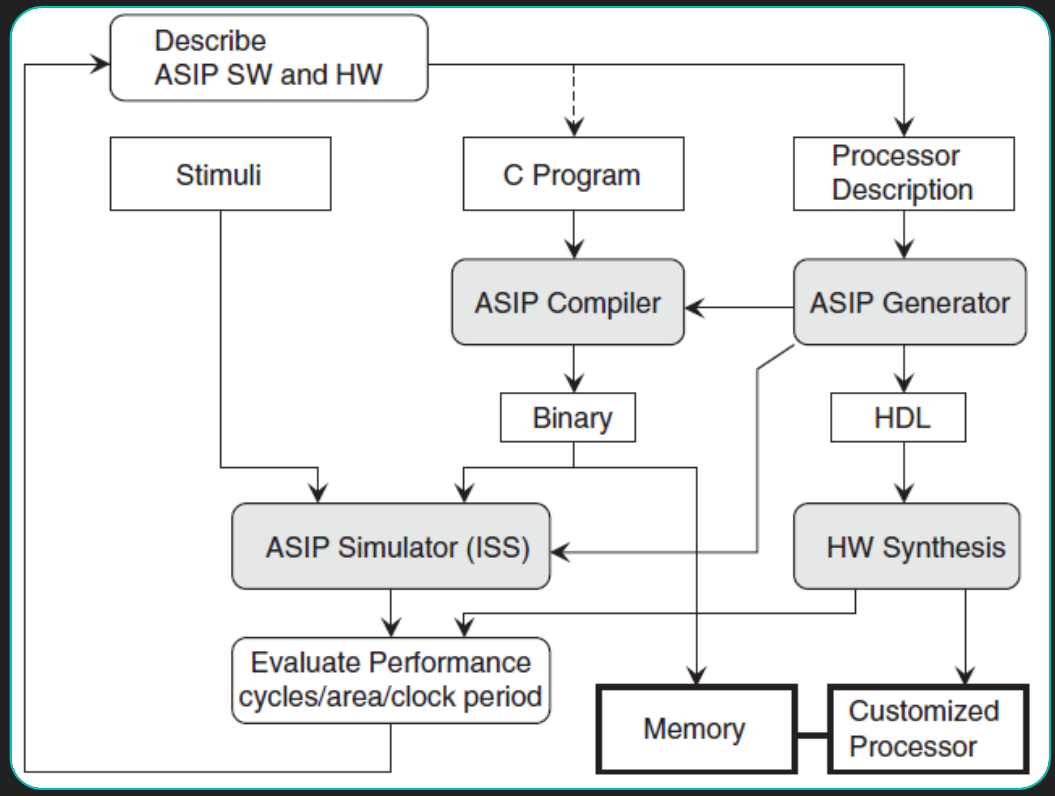


Interfaccia di coprocessore

- Quando si hanno trasferimenti di un'ingente quantità di dati, si preferisce progettare una interfaccia specificatamente creata per essere collegata alle porte ad essa dedicata del processore e gestita tramite istruzioni ad hoc. Queste componenti però risultano essere specifiche per processore, dopotutto nemmeno tutti i processi sono dotati di interfaccia di coprocessore.
- Il grado di riusabilità è limitato.
- Non vi sono vincoli di larghezza di parola, cosa che invece accade nelle interfacce mappate in memoria perché dipendenti dal bus.
- Latenza fissa.

ASIP

- L'integrazione HW/SW può essere accelerata in diversi metodi:
- Riservando una porzione dei codici operativi da un microprocessore per nuove istruzioni.
- Integrando I moduli personalizzati direttamente nell'architettura del microprocessore.
- Gestendo I moduli HW personalizzati usando delle nuove istruzioni.



ASIP

- Il set di istruzioni è definito dall'applicazione .
- Permette di automatizzare alcuni degli aspetti più ostici del codesign HW/SW.
- La progettazione con ASIP è incrementale.
- Inizialmente l'applicazione è un programma C, dopo averne valutato le performance si fanno degli aggiustamenti.
- La progettazione inizia con il programma, la descrizione del processore, la taglia dei registri, le istruzioni supportate e l'architettura della memoria.
- Successivamente l'ASIP generator creerà i componenti per l'asip e ciò include un kit di sviluppo e un sintetizzatore hardware.
- PRO: maggiore astrazione, riscontro veloce degli errori.
- CONTRO: Sequenziale quindi soggetto a colli di bottiglia, molto specifico.

Interfaccia custom nel NIOS II

- Il processore NIOS II possiede una interfaccia per istruzioni custom destinata a moduli HW e dotata di istruzioni predefinite.
- Supporta esecuzioni di lunghezza variabile attraverso un meccanismo di handshake a due vie.
- È permesso l'utilizzo di registri locali.