ESAME 2019

QUIZ:

1🡪8+4+2+1=15

2->D

3->C SPAZIO INDIRIZZAMENTO CPU=SPAZIO INDIRIZZAMENTO MEMORIA SE ISOLATED I/0

ALTRIMENTI SPAZIO INDIRIZZAMENTO CPU=SPAZIO INDIRIZZAMENTO MEMORIA + SPAZIO INDIRIZZAMENTO I/0

4->D

5->B

6-> 0010 0000 0101 0100 1010 0 100 0110-> 2+4+64=70

7->D

8->C

9->A

10-> lw $t0, (VAR1)

Sub $t0, $0, $t0

Sw $t0, ($VAR2)

9/10

ESERCIZI:

1)

Il dispositivo invia la richiesta all’Interrupt Controller il quale invia il segnale di Interrupt Request alla CPU. La CPU termina l’esecuzione dell’istruzione in corso, salva il contesto (ovvero Program Counter e Registro di stato) e invia il segnale di Interrupt Acknowledge. A questo punto l’IC scrive sul DBUS l’identificativo del dispositivo che ha fatto richiesta e la CPU, tramite tale codice, accede all’Interrupt Vector Table (che risiede in RAM) leggendo l’indirizzo dell’Interrupt Service Routine corrispondente, dando inizio al servizio. Al termine di quest’ultimo, la CPU ripristina il contesto e prosegue l’azione interrotta in precedenza.

2)

FETCH:

MAR<-PC

Y=0

CARRY=4

Z<-Y+PC+CARRY

WAIT RD

PC<-Z

WAIT MFC

MDR<-BUS ESTERNO

IR<-MDR

LW:

Y<-R3

BUS INTERNO<- CAMPO OPERANDO IR

Z<-Y+BUS INTERNO

MAR<-Z

WAIT RD

WAIT MFC

MDR<-BUS ESTERNO

R2<-MDR

3)

O=ac+a’c’+bd+b’d’

abcd o

0000 1

0001 1

0010 1

0011 0

0100 1

0101 1

0110 0

0111 1

1000 1

1001 0

1010 1

1011 1

1100 0

1101 1

1110 1

1111 1

4)

La memoria virtuale è un meccanismo utilizzato per usufruire di uno spazio di indirizzamento maggiore rispetto a quello realmente esistente dal punto di vista fisico. L’idea è quella di caricare soltanto una parte dell’immagine del programma in corso in memoria principale e il resto in memoria secondaria. Ciò favorisce il multitasking e l’indipendenza tra i programmi, oltre ad aumentare lo spazio di indirizzamento. Nella memoria virtuale, la CPU genera soltanto gli indirizzi logici, i quali vengono intercettati dalla MMU (posizionata tra la CPU e la CACHE) e tradotti in indirizzi fisici. Ciò avviene tramite la MAT (situata in memoria principale) ovvero la Memory Address Table; la MMU infatti lascia invariata la parte bassa (offset)dell’indirizzo logico, mentre quella alta viene sommata all’indirizzo di base della MAT per ottenere un nuovo indirizzo che farà parte (insieme all’offset precedente) dell’indirizzo fisico. A questo punto se l’indirizzo generato trova riscontro in memoria principale (o in cache), il blocco corrispondente viene fornito alla CPU, altrimenti il sistema va in PAGE FAULT e si richiede l’aiuto del sistema operativo che sposta il blocco richiesto dalla memoria secondaria alla memoria principale, in modo tale che possa essere fornito alla CPU.

ESAME 2021

QUIZ:

1->7

2->C

3->A

4->D

5->B

6->C

7->A

8->C

9->D

10->A SLT R1, R2, R3 METTE IN R1 1 SE R2<R3 ALTRIMENTI METTE 0

ESERCIZI:

1)

L’SSD è l’acronimo di Solid State Disk, ovvero un’unità di memoria a stato solido utile per essere sfruttata come memoria secondaria all’interno di un calcolatore elettronico. Un’SSD è costituita da un controller, per gestire le connessioni esterne, un’interfaccia per interagire con le altre unità e dei moduli di memoria FLASH NAND le quali si differenziano dalle NOR per resistenza ed efficienza. Una FLASH NAND si divide poi in MLC e SLC, le MLC memorizzano due bit per ogni cella e hanno una velocità di scrittura maggiore, le SLC invece memorizzano solo un bit per ogni cella ma sono più efficienti in lettura.

2)

Per gestire le temporizzazioni delle operazioni esistono due tipi di bus, sincroni e asincroni. I bus sincroni gestiscono le tempistiche delle operazioni tramite la condivisione di un colpo di clock tra master e slave oppure l’utilizzo di colpi di clock diversi ma alla stessa frequenza, la quale è imposta dal dispositivo a frequenza minore. I bus sincroni sono ottimi per le trasmissioni a limitate distanze. I bus asincroni invece gestiscono le temporizzazioni tramite l’invio di segnali asincroni, il cui insieme è detto handshaking. Ciò permette alle trasmissioni di essere più veloci e ottimizzate per distanze elevate. Dal momento in cui possono esistere più master del bus, bisogna gestire i conflitti tra di essi, dato che solo uno di loro per volta può utilizzare il bus. Ciò avviene tramite due tipi di arbitraggio, distribuito e centralizzato. L’arbitraggio distribuito SCSI è costituito da 8 linee a cui sono collegati i dispositivi master, ogni linea ha una priorità che è crescente con l’indice (priorità crescente da 0 a 7) della linea stessa. Inoltre è presente un’ulteriore linea, quella di bus busy, la quale appena viene disattivata, segnala la possibilità di prendere il bus da parte del master con priorità più alta. A questo punto infatti tutti i dispositivi alzano la propria linea, e quello con priorità più alta prende possesso del bus. L’arbitraggio centralizzato invece fa riferimento ad un arbitro e si divide in daisy chaining, polling e richieste indipendenti. Nel caso del daisy chaining abbiamo sempre tre segnali di controllo, ovvero Grant, Busy e Request. Il segnale di Busy è un segnale che va in unput all’arbitro ed è condiviso da tutti i dipositivi, esso viene utilizzato per segnalare l’occupazione del bus, quello di Request è condiviso ugualmente, anch’esso in input all’arbitro e segnala la richiesta di possesso del bus e infine abbiamo il segnale di Grant in uscita dall’arbitro e in ingresso al primo elemento della catena di dispositivi, esso viene propagato per tutta la catena finchè non giunge al primo dispositivo che aveva fatto richiesta. Ha come vantaggio il basso costo dovuto ai pochi segnali e come svantaggi la priorità statica dovuta alla catena e l’intollerabilità ai guasti. Il polling utilizza 2+logn linee, dove n sono i dispositivi, che sono usate come Grant, in cui viene inserito il codice del dispositivo con priorità più alta da parte del poll counter. L’operazione viene eseguita finchè non si inserisce il codice di un dispositivo che ha fatto richiesta, il quale prenderà il bus. Le altre due linee sono quella di busy e request con funzionamento analogo al caso precedente. I vantaggi sono la priorità dinamica grazie al poll counter e la tollerabilità ai guasti, lo svantaggio è il costo maggiore dovuto al numero di linee. Infine abbiamo le richieste indipendenti in cui ci sono 2n+1 linee, con n il numero di dispositivi. Ogni dispositivo infatti pilota i segnali di request e grant, mentre il segnale di busy è condiviso. Questo permette la massima flessibilità in termini di priorità perché l’arbitro riconosce immediatamente il dispositivo che ha fatto richiesta e gli cede il bus, se questo è libero. I vantaggi sono la tollerabilità ai guasti e la priorità dinamica mentre lo svantaggio è rappresentato dall’elevato costo dovuto al numero di linee.

3)

L’unità di controllo microprogrammata è un’unita caratterizzata dall’elevato costo unitario, ma dal basso costo di progetto, presenta infatti un’alta flessibilità che la rende preferibile rispetto all’unità di controllo cablata, d’altro canto però è meno veloce ed è più costosa. La sua architettura è costituita da un circuito di generazione dell’indirizzo che è connesso all’IR del calcolatore, da cui legge l’istruzione in corso. Esso pilota un segnale in output al microPC il quale accede alla memoria in microcodice accedendo alla microistruzione corrispondente, che a sua volta pilota i segnali di controllo corrispondenti nel microIR. Infine il microIR pilota tali segnali verso l’unità di elaborazione. L’unità di elaborazione comunica direttamente anche con la circuiteria di generazione dell’indirizzo interna all’unità microprogrammata.

In questo caso il microIR è costituito da 56 bit, tanti quanti sono quelli posseduti da una parola.

4)

cba o1 o2

000 1 0

001 1 0

010 1 0

011 0 0

100 0 0

101 0 0

110 0 1

111 0 1

O1= a’c’+c’b’

O2= cb