

Computersysteme



Eine frühere Klausuraufgabe zur Übung

Eine frühere Klausuraufgabe zur Übung:

Schreiben Sie ein Assembler-Programm, das von einer Folge von n 32-Bit-Zahlen im Zweier-Komplement die Anzahl derjenigen Zahlen ermittelt, die durch 4 aber nicht durch 8 teilbar sind. Diese Anzahl soll an die Adresse 1512 in den Speicher geschrieben werden. Die Zahl n ($1 \leq n \leq 100$) steht am Anfang des Programms an Adresse 1000 im Speicher. Die n Zahlen stehen an den Adressen 1004 bis $1000 + 4n$.

- Schreiben Sie das Assembler-Programm und verwenden Sie dabei die anliegenden DLX-Befehle.
 - Beschreiben Sie die Arbeitsweise Ihres Programms und geben Sie die Registerbelegung an.
 - Kommentieren Sie jede Programmzeile.
-
- Genau lesen, Stichpunkte machen .
 - Kleines Beispiel
 - Zahlenformat, Grenzfälle, Probleme?
 - Registerbelegung, Algorithmus (z.B. als Flussdiagramm)
 - Kleine Codestücke schreiben und testen (was ist schwierig?)
 - Programm kommentieren und beschreiben
 - Programm schrittweise durchlaufen lassen
 - Zum Schluss auch größere Beispiele testen.

Schreiben Sie ein Assembler-Programm, das von einer Folge von n 32-Bit-Zahlen im Zweier-Komplement die Anzahl derjenigen Zahlen ermittelt, die durch 4 aber nicht durch 8 teilbar sind. Diese Anzahl soll an die Adresse 1512 in den Speicher geschrieben werden. Die Zahl n ($1 \leq n \leq 100$) steht am Anfang des Programms an Adresse 1000 im Speicher. Die n Zahlen stehen an den Adressen 1004 bis $1000 + 4n$.

Instr.	Description	Instr.	Description
ADD	add	SLL	shift left logical
ADDI	add immediate	SLLI	shift left logical immediate
AND	and	SLT	set if less than
ANDI	and immediate	SLTI	set if less than immediate
BEQZ	branch if equal to zero	SNE	set if not equal
BNEZ	branch if not equal to zero	SNEI	set if not equal to immediate
J	jump	SRA	shift right arithmetic
JAL	jump and link	SRAI	shift right arithmetic immediate
JALR	jump and link register	SRL	shift right logical
JR	jump register	SRLI	shift right logical immediate
LHI	load high bits	SUB	subtract
LW	load word	SUBI	subtract immediate
OR	or	SW	store word
ORI	or immediate	XOR	exclusive or
SEQ	set if equal	XORI	exclusive or immediate
SEQI	set if equal to immediate		
SLE	set if less than or equal		
SLEI	set if less than or equal to immediate		

/Programmbeschreibung s. Extrablatt

/Registerbelegung:

/R1: Pointer auf Speicher, zunächst $4*n$, wird in 4er-Schritten reduziert.

/R2: Aktuell zu testende Zahl aus Speicher

/R3: Anzahl der durch 4 aber nicht durch 8 teilbaren Zahlen

/R4: Hilfsvariable für Teilbarkeitstest

LW R1, 1000(R0) /Speichere n in R1

SLLI R1, R1, #2 /Multipliziere R1 mit 4. R1 ist Pointer auf Speicher

ADD R3, R0, R0 /Initialisiere Anzahl der gefundenen Zahlen mit 0

Loop: /Schleifenbeginn

LW R2, 1000(R1) /Hole zu prüfende Zahl aus Speicher und speichere diese in R2

ANDI R4, R2, #7 /Prüf, ob 8 teilt R2.

BEQZ R4, LoopEnd / 8 teilt R2 => R2 wird nicht berücksichtigt

ANDI R4, R2, #3 /Prüf, ob 4 teilt R2.

BNEZ R4, LoopEnd / 4 teilt nicht R2 => R2 wird nicht berücksichtigt

ADDI R3, R3, #1 /Erhöhe Anzahl der gefundenen Zahlen um 1

LoopEnd: /Schleifenende

SUBI R1, R1, #4 /Reduziere Pointer auf Speicher um 4

BNEZ R1, Loop /Solange Pointer nicht auf 0, springe zum Schleifenbeginn

SW 1512(R0),R3 /Speicher Anzahl der gefundenen Zahlen an Speicheradresse 1512

HALT