Übung zur Vorlesung Technische Grundlagen der Informatik

Prof. Dr. Andreas Koch
Thorsten Wink

TECHNISCHE UNIVERSITÄT DARMSTADT

Wintersemester 09/10
Übungsblatt 11 - Lösungsvorschlag

Als Simulator für MIPS-Assembler wird der MARS-Simulator empfohlen. Sie können ihn unter http://courses.missouristate.edu/KenVollmar/MARS/ herunterladen. Er ist auch auf den RBG-Poolrechnern installiert.

Aufgabe 11.1 MIPS-Assembler

Wandeln Sie den folgenden Hochsprachen-Code in MIPS-Assemblercode um. Die Variablen a, b, c sind in den Registern $s0, $s1, $s2 abgelegt.

- a = a + b;
  ```
  add $s0, $s0, $s1
  ```
- a = 23;
  ```
  addi $s0, $zero, 23
  li $s0, 23
  ```
- $a = b^4 + (c+1)b^2 + 1$;

  ```
  addi $t0, $s2, 1   # Berechne c + 1
  mul $s1, $s1 $s1   # Quadriere b -> b^2
  mul $s0, $t0, $t1  # Multipliziere c + 1 mit b^2 in a
  mul $t1, $t1, $t1  # Quadriere b^2 -> b^4
  add $s0, $s0, $t1  # Addiere b^4 zum Zwischenergebnis
  addi $s0, $s0, 1   # Addiere 1 zum Zwischenergebnis, fertig
  ```
- a = feld[5];

  ```
  addi $t0, $zero, 20 # Offset ist 5 * 4
  lw $s0, feld($t0)   # Lade feld[5]
  ```
- feld[a]++;

  ```
  sll $t0, $s0, 2    # Offset ist a * 4
  lw $t1, feld($t0)  # Lade feld[a]
  addi $t1, $t1, 1   # Inkrementiere feld[a]
  sw $t1, feld($t0)  # Speichere feld[a] zurück
  ```
- ```
  c = 0;
  while (a > 0) {
    a = a-b;
    c++;
  }
  ```
  ```

```
add $s2, $zero, $zero     # c = 0
while:
  blez $s0, endwhile      # Verlasse Schleife falls a <= 0
  sub $s0, $s0, $s1       # a = a-b
  addi $s2, $s2, 1        # c++
  j while                 # Wiederhole Schleife
endwhile:
```

- ```
  a = 0;
  for (b = 0; b < 1024; b++) {
    a = a + feld[b];
  }
  ```

  ```
  add $s0, $zero, $zero   # a = 0
  add $s1, $zero, $zero   # b = 0
  addi $t1, $zero, 4096   # Für Vergleich in Schleife
  loop:
    lw $t0, feld($s1)     # Lade feld[b]
    add $s0, $s0, $t0     # a = a + feld[b]
    addi $s1, $s1, 4      # b++ (als Wort-Offset)
    blt $s1, $t1, loop    # Wiederhole Schleife falls b < 4096
  ```

Aufgabe 11.2 MIPS-Befehlsformate

Welche Befehlsformate gibt es beim MIPS-Prozessor? Worin unterscheiden sie sich?
Siehe Folien 6-31 bis 6-37.
  ```

Aufgabe 11.3 Disassemblierung

Gegeben ist der folgende Auszug aus dem MIPS-Befehlsspeicher. Geben Sie den zu diesem Programm gehörenden MIPS-Assemblercode an. Die erste Spalte enthält die Adresse, die zweite enthält den Befehl. Was berechnet das Programm?

```
0x00400000 0x20080000
0x00400004 0x20090001
0x00400008 0x0089502a
0x0040000c 0x15400003
0x00400010 0x01094020
0x00400014 0x21290002
0x00400018 0x08100002
0x0040001c 0x01001020
```

```
0x00400000 0x20080000  #addi $t0, $zero, 0
0x00400004 0x20090001  #addi $t1, $zero, 1
0x00400008 0x0089502a  #loop: slt $t2, $a0, $t1
0x0040000c 0x15400003  #  bne $t2, $zero, finish
0x00400010 0x01094020  #  add $t0, $t0, $t1
0x00400014 0x21290002  #  addi $t1, $t1, 2
0x00400018 0x08100002  #  j loop
0x0040001c 0x01001020  #finish: add $v0, $t0, $zero
```

Das Programm summiert die ungeraden Zahlen von 0 bis zu dem wert, der in $a0 übergeben wird. Das Ergebnis steht in $v0.

Aufgabe 11.4 MIPS-Registersatz in Verilog

Der MIPS-Prozessor besitzt einen Registersatz mit 32 Registern zu je 32 Bit. Er hat einen Drei-Port-Speicher mit zwei Lese-Ports und einem Schreib-Port. Implementieren Sie den Registersatz in Verilog.

```
module register(
  input         clk,
  input         write_enable,
  input  [4:0]  read_addr1, read_addr2, write_addr,
  input  [31:0] write_data,
  output [31:0] read_data1, read_data2
  );

  //Registerspeicher
  reg [31:0] RAM[31:0];

  //Schreiboperation, Taktsynchron
  //auf Register 0 kein Schreiben erlaubt
  always @(posedge clk)
    if (write_enable & write_addr != 0)
      RAM[write_addr] <= write_data;

  //Leseports
  assign read_data1 = RAM[read_addr1];
  assign read_data2 = RAM[read_addr2];
endmodule
```

Plagiarismus

Der Fachbereich Informatik misst der Einhaltung der Grundregeln der wissenschaftlichen Ethik großen Wert bei. Zu diesen gehört auch die strikte Verfolgung von Plagiarismus. Weitere Infos unter www.informatik.tu-darmstadt.de/plagiarism