

6. Adressierungsarten

6.1 Register Adressing

Register Adressing

Der für die Instruction benötigte Operand steht in einem der 32 Register des Prozessors.

6.2 Immediate Adressing

IAderseeing

Der Operand ist eine Konstante im Befehl. Diese Adressierungsart benutzt man um bei Verwendung von kleinen Konstanten, Speicherzugriffe zu vermeiden.

6.3 Base Adressing

Base Adressing

Bei dieser Art von Adressierung befinden sich die benötigten Operanden im Speicher oder kommen von irgendeinem Eingabegerät.

Die Adresse des Operanden ist die Summe der im zweiten Register gespeicherten Basis-Adresse und der Konstanten, die in den hinteren 16 bit des Befehls steht.

7. Subroutinen:

Eine Subroutine (procedure call) ist ein Programmierwerkzeug. Kleine Programme, die man oft braucht, werden in einem bestimmte Speicherbereich abgespeichert und können dann aufgerufen werden. Sie werden aufgerufen mit dem Befehl Jump and link

jal <Adresse der Subroutine>

Am Ende der Subroutine steht der Befehl Jump return

jr \$ra

Das Register ra ist eines der 32 Register des MIPS, in dem bei dem Befehl jal die Adresse des im Programm folgenden Befehls gespeichert wird.

Wenn die Subroutine an ganz anderer Stelle noch mal gebraucht wird, muss sie trotzdem nur einmal programmiert werden.

d.H. bei einem etwas umfangreicheren Programm, sollte man auf jeden Fall Subroutinen programmieren. Das eigentliche Programm wird dadurch übersichtlicher und wenn eine Subroutine von mehreren Stellen im Programm benutzt wird spart man Speicherplatz.

Um zu vermeiden daß im Verlauf einer Subroutine Inhalte von Registern, die an anderer Stelle noch gebraucht werden, überschrieben werden, speichert man diese vor Beginn einer Subroutine.

Den Vorgang, vor Beginn einer Subroutine Daten aus den Registern des Prozessors in den Speicher wegzusichern nennt man push.

Nach Beendigung der Subroutine Daten wiederholen nennt man pop.

Es gibt eine Festlegung welche Register zu Beginn jeder Subroutine gesichert werden und welche nicht.

Liste 138

Es gibt ein Register das dafür da ist, festzulegen und zu speichern wo (an welchen Speicheradressen) die Daten gespeichert werden.

Das Register heißt \$sp (Stack Pointer).

8 Von der Hochsprache zur Machinensprache.

Folie A%C 156

8.1 Compiler

Man programmiert in Hochsprachen, weil es leichter und schneller ist, und mittlerweile Compiler sehr effektiv übersetzen.

Ein Compiler ist ein Programm das Programme von einer Hochsprache in ein Assembler Programm übersetzt.

Mit dem entsprechenden Compiler sind Programme also universell einsetzbar und nicht mehr an den Prozessor gebunden, in dessen Befehlsatz sie programmiert wurden.

8.2 Assembler

der Assembler übersetzt die Befehle und Registerbezeichnungen in Maschinensprache.

8.3 Linker

Werden in den zu übersetzenden Programm irgendwelche Subroutinen verwendet die nicht Teil des Programms sind, sondern irgendwo anders gespeichert sind, muß die Verbindung zu diesen hergestellt werden.

Diese Aufgabe erfüllt der Linker (Binder).

8.4 Loader

Der Lader erhält von der Speicherverwaltung des Betriebssystem die Anweisung wo das Programm gespeichert werden soll und rechnet die Adressen des Programms in die entsprechenden Adressen des Speichers um.

9. Zahlen und Buchstaben

Buchstaben können durch Zahlen dargestellt werden. (codieren)

Der wichtigste Zahlencode zur Darstellung von Buchstaben ist der ASCII-Code.

ASCII: American-Standard-Code for Information-Interchange

Der ASCII-Code umfasst 128 Zeichen:

- große und kleine Buchstaben (52 Zeichen)
- Ziffern von 0-9 (10 Zeichen)
- Sonderzeichen und Steuerzeichen (66 Zeichen)

Folie ASCII-Code

Es gibt noch eine erweiterte Form des ASCII Codes, die auch Sonderzeichen wie Ä enthält, die dann 256 Zeichen umfasst und entsprechend 8 bit benötigt.