

Mikroprozessortechnik - Das verständliche Communityskript

Übung: Assembler



Aufgabe: Der *CORDIC*-Algorithmus

Der CORDIC-Algorithmus (**C**oordinate **R**otation **D**igital **C**omputer) wurde Ende der 50er Jahre als effizienter Algorithmus zur Berechnung von trigonometrischen Funktionen entwickelt. Zuvor erfolgte die Berechnung aufwändig durch Taylorapproximation oder Lookup-Tables mit mittelmäßig guter Konvergenz und numerischer Stabilität. Wir möchten den CORDIC-Algorithmus verwenden, um Funktionswerte trigonometrischer Funktionen \sin , \cos , \tan und \cot zu berechnen. Hierzu wurde dieser Algorithmus in zweidimensionaler Form als eine Funktion in der MIPS-Assemblersprache implementiert. Es soll nun mithilfe des CORDIC-Algorithmus die Funktionsgraphen der besagten trigonometrischen Funktionen auf einem Display dargestellt werden.

- Die Assembler-Vorlage kannst Du Dir hier: <http://cpumuseum.jimdo.com/museum/architectures/mips/> herunterladen. Kopiere den Inhalt der PDF in den MARS-Editor und speichere den Code unter dem Namen `cordic.asm`.
- Öffne `cordic.asm`. Der MIPS-Assembler-Simulator Mars erlaubt die Anbindung eines Grafikdisplays. Gehe hierzu auf **Tools** und anschließend auf **Bitmap Display**. Wir verwenden für die Übung die voreingestellte Displayauflösung von 512 x 256
- Das Tool färbt die Pixel eines Bitmaps entsprechend dem Inhalt einer Speicherzelle ein. Der Inhalt wird als 24-Bit RGB-Farbwert interpretiert mit Bits 16-23 für Rot, Bits 8-15 Grün und Bits 0-7 für Blau. Es ergeben sich für die Farben:

Weiß = 0xFFFFFFFF

Rot = 0xFFFF0000

Grün = 0xFF00FF00

Blau = 0xFF0000FF

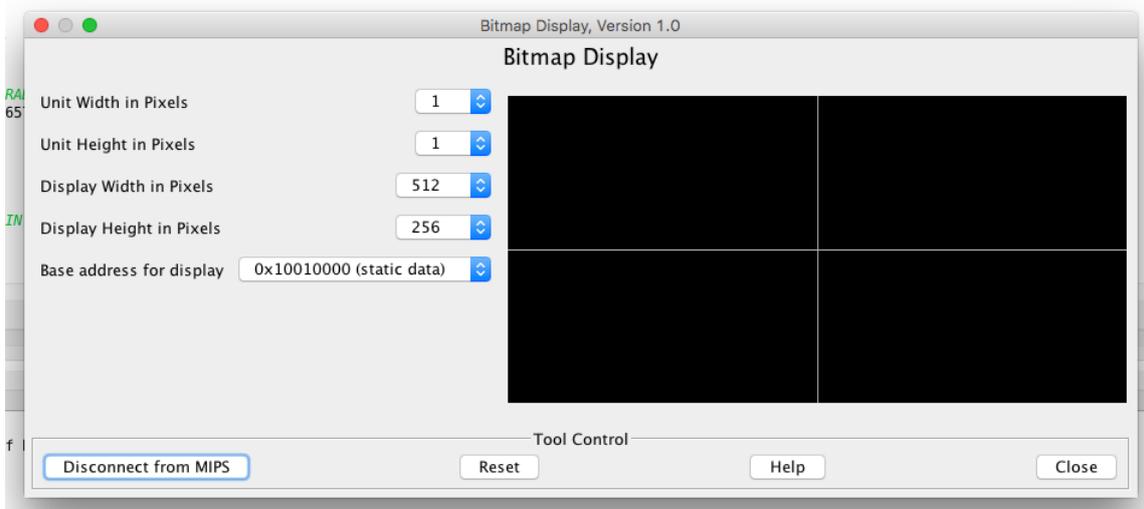


Abbildung 1: Bitmap Display mit eingezeichnetem Koordinatenkreuz

- Klicke auf **Connect to MIPS**, anschließend im Simulator auf das Werkzeugsymbol (Assemble) auf das 'Play'-Symbol (Debug), um den Code auszuführen
 - Nach dem Ausführen sollte nun ein leeres Koordinatensystem zu sehen sein. Auf diesem Koordinatensystem sollen nun (zunächst) die Funktionsgraphen von Sinus und Kosinus von $-\pi/2$ bis $\pi/2$ aufgetragen werden
- (a) Implementiere zunächst eine Funktion `draw(x, y, color)`, die ein Pixel auf dem Bitmap-Display an der Position (x, y) mit der Farbe `color` färbt.
- `$a0`: x-Koordinate zwischen 0 und 511
 - `$a1`: y-Koordinate zwischen 0 und 255
 - `$a2`: Farbwert (`color`) zwischen `0x00000000` und `0xFFFFFFFF`

Die **Start**adresse des Displays kann mit dem Befehl `la $reg, graph` geladen werden. Der Adressraum des Displays sei wie folgt im Arbeitsspeicher abgelegt:

graph + 524284	graph[255][511]
graph + 524280	graph[255][510]
...	...
...	...
graph + 2052	graph[1][1]
graph + 2048	graph[1][0]
...	...
...	...
graph + 8	graph[0][2]
graph + 4	graph[0][1]
graph	graph[0][0]

Abbildung 2: Adressraum für das Bitmap-Display

(b) Realisiere eine Funktion `plot(iterations, function)`, die als Parameter die Anzahl der Iterationen, sowie den gewünschten Funktionstyp (sin oder cos) übergeben bekommt:

- `$a0`: Anzahl der Iterationen. Fester Wert größer gleich null. Der Inhalt in `$a0` darf nicht verändert werden und wird an eine Unterfunktion weitergeleitet
- `$a1`: Gewünschte Funktion, kann zwei Werte annehmen: 0: $\cos(x)$, 1: $\sin(x)$. Der Inhalt in `$a1` darf nicht verändert werden und wird an ein Unterfunktion weitergeleitet

Die Funktion `plot` soll die Funktionswerte von CORDIC im Bereich von $-\pi/2$ bis $\pi/2$ auf dem Bitmap-Display grafisch plotten. Verwende folgende Unterfunktionen zur Realisierung der Funktion:

`cordic`. Erfordert drei Übergabeparameter und gibt einen Wert zurück.

- Übergabeparameter `$a0`: Anzahl der Iterationen
- Übergabeparameter `$a1`: Gewünschte Funktion: 0: $\cos(x)$, 1: $\sin(x)$, 2: $\tan(x)$, 3: $\cot(x)$
- Übergabeparameter `$f12`: Winkel (Typ `double`). Muss in der Aufgabe **nicht** berücksichtigt werden. Wird automatisch verwaltet durch `graph_initialize` und `graph_update` (siehe unten) und wird inhaltlich nicht behandelt, da Gleitkommaoperationen enthalten sind. Nimmt Werte zwischen $-\pi/2$ und $\pi/2$
- Rückgabewert `$v0`: Der eigentliche CORDIC-Algorithmus (für sin und cos) liefert `double` Gleitkommawerte zwischen -1 und 1. Für den Plot wird dieser Bereich auf 64 bis 192 skaliert und in eine Ganzzahl umgewandelt. Der tatsächliche Wertebereich für den Plot erstreckt sich also im Bereich zwischen -2 und 2.

`graph_initialize`. Initialisiert CORDIC: Legt den Startwinkel $-\pi/2$ fest. (`$f12 = -\pi/2`)

`graph_update`. Addiert $\pi/512$ auf den aktuellen Wert in `$f12`. (`$f12 += \pi/512`)

Es soll die Farbe Rot für beide Graphen verwendet werden.

Hinweis: Verwende ausschließlich `$s`-Register. Beachte, dass hier Unterfunktionen aufgerufen werden!

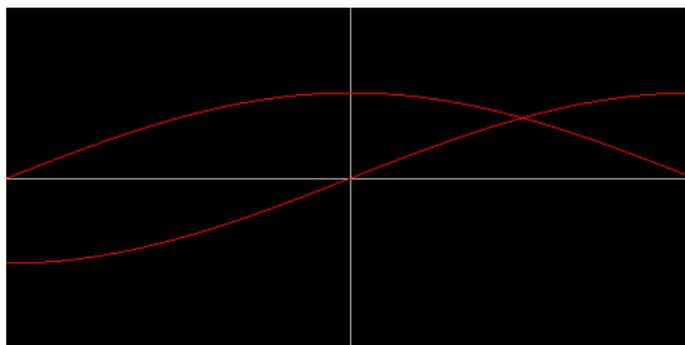


Abbildung 3: Die Ausgabe am Display nach erfolgreicher Bearbeitung von (a) und (b)

(c) Die in (b) geschriebene Funktion `plot` soll erweitert werden und nun den gewünschten Funktionstyp auswerten und dem passend eine Farbe zuordnen.

- Sinus: Grün
- Cosinus: Rot
- Tangens: Blau
- Cotangens: Gelb (*Tipp*: Gelb = Rot + Grün)

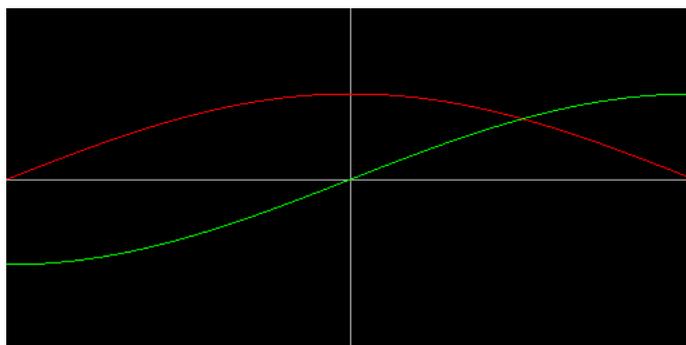


Abbildung 4: Die Ausgabe am Display nach erfolgreicher Bearbeitung von (c)

- (d) Bisher sind wir davon ausgegangen, dass unser Übergabeparameter `$a1` nur den Wert 0 (cos) oder 1 (sin) annehmen kann. Nun kann `$a1` auch den Wert 2 (tan) oder 3 (cot) übergeben bekommen. Unser Befehl `plot` kann also auch die Anweisung erhalten, Tangens oder Cotangens zu plotten. Hierzu ergibt sich folgendes Problem:

Der Tangens/Cotangens ist für bestimmte Werte undefiniert und strebt in dessen Umgebung Werte bis $\pm\infty$ an. Der skalierte Rückgabewert (`$v0`) kann also auch sehr große Zahlen jenseits des Bereichs zwischen -2 und 2 annehmen. Würden wir diesen als y-Wert dem `draw`-Befehl übergeben, würde dieser in Adressbereichen außerhalb des Displayadressbereichs schreiben.

Die Funktion `plot` muss also um eine Einheit erweitert werden, die erkennt, dass der Graph vom Tangens oder Cotangens geplottet werden soll. Liegt der Rückgabewert von CORDIC außerhalb des plotbaren Bereichs, soll kein Plot erfolgen (`draw` muss übersprungen werden).

Zum Testen der Funktionalität von der Tangens/Cotangens-Erweiterung kannst Du in die `main`-Funktion scrollen. Dort ist der Tangens/Cotangens-Plot auskommentiert. Entferne dort die Kommentarkennzeichnung.

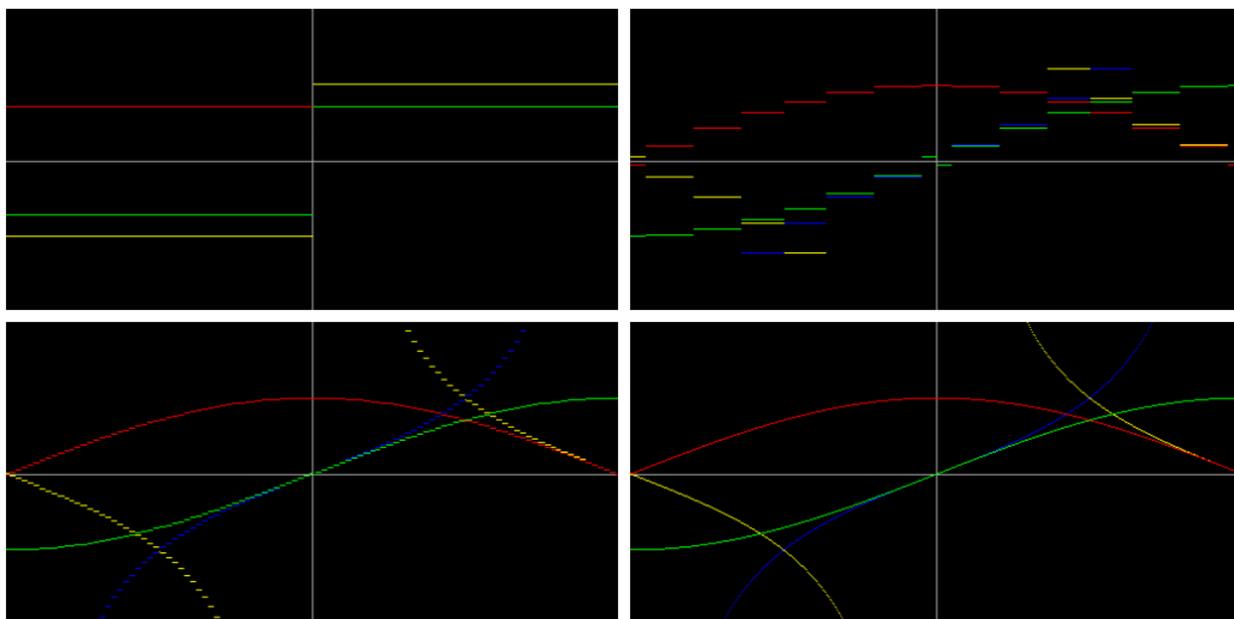


Abbildung 5: Die Ausgabe am Display nach erfolgreicher Bearbeitung von (d). CORDIC Grapher jeweils für 1, 4, 7 und 30 Iterationsschritte ausgeführt