Beginn: 13:00
Ort: Arbeitsgruppenraum C2.227
Anwesende: alle Projektgruppenteilnehmer mit Außnahme von Franz Rammig.
Als einzige Punkte der Tagesordung lagen vor:

1. Organisatorisches
2. Besprechung der Anforderungen für das P30-Papier bezüglich der einzelnen Teilbereiche.

Zuerst stellte die Gruppe 'Rechnenkern' ihr bisheriges Konzept vor:
Es sollen maximal 15 Stützstellen verwendet werden. Bei mehreren Werten müssen die Werte dann sequentiell geladen werden. Gerechnet werde dann auf den 32 32-Bit Registern, die gemäß Spezifikation von der Architektur vorgegeben sind. Die Berechnungen laufen dann auf einer 32-Bit ALU. Im Punkt der Funktionen wurden dann die angegebenen Funktionen von den vier Grundrechenartein auf Vergleiche und komplexere Funktionen erweitert. Als nächster Punkt wurde der Speicherbedarf für die Berechnungen diskutiert. Man benötigt für n Stützstellen eine $(n-1)$ x $(n-2)$ Martix zur Berechnung der y_i. Da bei n Stützstellen $n-1$ Polynome gebildet werden können; $n-2$, weil $y_0^{''} = y_n^{''} = 0$ ist und daher nicht gespeichert werden muß. Betrachtet man diese Überlegung zusammen mit der zuvor getroffen Beschränkung auf max. 15 Stützstellen so erhält man: es werden 14 x 13 x 4 Byte an Speicher (= 728 Bytes) benötigt um die Koeffezienten der Polynome zu speichern. Für die Berechnung der Punkte werden 2 x 16 x 4 Byte (= 126 Bytes) gebraucht. Daraus ergibt sicht ein gesamter Speicher- bzw. Registerbedarf von 856 Bytes. Es wurde dann die Möglichkeit diskutiert diesen Speicher auf dem Chip direkt in Form von Registern oder Speicherzellen anzulegen. Es wäre prinzipiell möglich beliebig viele Speicher / Register anzulegen, es bleiben aber nur 32 32-Bit Register an außenhin 'sichtbar. Dabei wurde auch gleich der Bedarf an Gattern auf dem Simulator diskutiert und für zunächst ausreichend befunden. Als nächstes stellte die Gruppe einen möglichen Programmablauf vor:

- Stützstellen von der CPU anfordern
  - Anzahl der Stützstellen in Register No. 0
  - Stützstellen in Register No. 1, 2, .. n (mit n = Anzahl der Stützstellen)
- Startsignal, mir dem die eigentlich Splineberechnung gestartet wird
- Splineinterpolation
  - Berechne Koeffezientender $y_i^{''}$ und speichere sie in der Matix ab
  - Löse Gleichungssystem nach der Gauß'schen Methode
  - Berechne alle a_i, b_i, c_i und d_i aus den berechneten $y_i^{''}$
- Ausgabe
  - Kopiere alle n-1 Koeffizienten dritten Grades in die Register
  - warte bis die Werte in den Speicher übernommen worden sind
  - Schicke die anderen Koeffezienten zweiten, ersten und nullten Grades in gleicher Weise

Im wesentlichen wurde keine Kritik an dem Algorithmus geäußert, es wurde aber zu bedenken gegeben, daß es vielleicht sinnvoller ist, die Koeffizienten polinomweise zurück zu liefern, da es somit schon einer Anwendung möglich wäre die Daten zu verwenden. Zu eventuellen Problemen mit der Beschränkung der Register, die nach 'außen sichtbar' sind, stellten wir fest, daß falls mehr als 32 Register benötigt werden, ein Register zum Umschalten auf andere Registersätze reserviert bleiben muß. Somit stehen dann nur 31 Register zur Verfügung. Auch gabe es zu Bedenken, daß nicht der Coprozessor der Master ist, der den Berechnungsablauf steuert, sondern daß die Integerunit alle Rechenvorgänge regelt. Die Idee mit dem Startsignal läßt sich gut realsieren. Des weitern soll die Gruppe sich überlegen, welche Signale sie vom Interface erwartet.
Als nächste Gruppe stellte die Gruppe 'Benutzungsschnittstelle' weitere Aspekte zu ihrem Konzept dar. Der bisherige Stand umfaßt 8 Befehle:

- SplAddPt (x,y) - Punkt hinzufügen
- SplDelPt (x,y) - Punkt löschen
- SplClrPts - alle Punkte löschen
- SplCalc - Spline berechnen
- SplGetPt (nr) - Koordinate des Punktes 'nr' liefern
- SplGetCoeff (nr, a, b, c, d) - Polynomkoeffezienten liefern
- SplGetRange (nr, x_1, y_1, x_2, y_2) - Definitionsbereich für Polynom liefern
- SplErrText (nr) - textuelle Fehlermeldung liefern

Zum Befehlssatz wurde gesagt, daß damit alle von marktüblicher Software gebotene Optionen abgedeckt sind. Desweitern wurde die Liste der Fehlermeldungen nicht verändert:

Fehlernummer	Bedeutung
0	Ok
1	Punkt bereits vorhanden
2	Koordinate bereits vergeben
3	Maximale Anzahl an Pukten überschritten
4	zu wenig Speicherplatz
5	Punkt nicht vorhanden
6	zu wenig Punkte
7	Spline nicht berechnet
8	Splinepolynomnummer nicht verwendent

Die Gruppe regte ebenfalls an, sich an eine Beschränkung auf 15 Stützstellen zu einigen, da sich die natürlichen Splines nicht kaskadieren lassen. Mit dem verwendeten Algorithmus werden keine weitergehende Bestimmungen über das Verhalten der Splines an den Endpunkten getroffen. Es sich daher möglich, daß es zu unstetigen Stellen kommt. Diese sind aber für Splines nicht zulässig (siehe Skizze). Eine Möglichkeit diesen Mißstand zu beseitigen wäre, die Steigung an den Endpunkten zu speichern und als Kriterium für die Berechnung des nächsten Polynoms hinzu zu ziehen. Als vorläufigen Beschluß wurde vereinbart, daß wir auf ein Kaskadierung zunächst verzichen, und diese eventuell später zu erweitern.

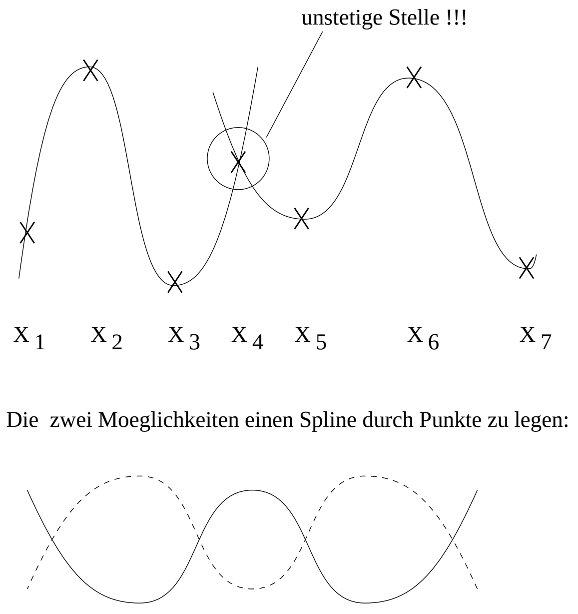

Abbildung 1: unstetige Stelle

Danach folgten die Ausführungen der Gruppe Betriebssystem. Zuerst wurde berichtet, daß die FPU über ein Bit im Statusregister der IU geprüft wird. Eventuelle Rückmeldungen erfolgen über sogenannte TRAPs. Die weiteren Ausführungen sind:

- FPU und Coprozessor werden durch Maschinenroutinen angesteuert. Jede Maschinenroutine besitzt ein Bitfeld, in dem angegeben ist, wer den Befehl auszuführen hat : IU, FPU oder Coprozessor.
- Es existiert ein Statusregister, in dem angegeben werden kann, ob die FPU existiert oder nicht.
- Wird die FPU durch einen Eintrag im Statusregister ausgeschaltet, so werden Befehle, die den Vermerk 'FPU' besitzen, emuliert. Für jeden durch die FPU ausführbaren Befehl steht deshalb bereits eine entsprechende Maschinenroutine zur Verfügung.
- Ist die FPU als angeschlossen deklariert, und soll ein Befehl, welcher als FPU–Befehl markiert ist, bearbeitet werden, so wird dieser Befehl an die FPU weitergeleitet. Falls es irgendwelche Probleme mit diesem Befehl gibt, meldet die FPU ein TRAP–Signal zurück.
- Es gibt verschiedene Typen von TRAPS, je nachdem, wann und wo sie aufgetreten sind:
  - Kernel exception : Kernel $\longrightarrow$ User Befehl wurde nicht ausgeführt
  - Unfinished FPO
  - Unimplemented FPO
- Werden Coprozessor und FPU zusammen eingebaut, dann gibt es Probleme:
  - Beide werden parallel eingebaut
    $\longrightarrow$ beide erhalten gleichzeitig die selben Befehle

$\longrightarrow$ wenn die FPU einen unserer neuen Befehle erhält, so gibt er ein TRAP zurück, während unser Coprozessor unseren Befehl schon bearbeitet
$\longrightarrow$ Probleme mit dem TRAP–handling

  – Beide werden seriell eingebaut: $\longrightarrow$ unser Coprozessor muß alle Befehle, die er nicht kennt, weiterleiten, auch wenn er gerade mit der Abarbeitung einer Splineberechnung beschäftigt ist $\longrightarrow$ Probleme bei der Implementation unseres Chips

- Wird unser Coprozessor statt der FPU eingesetzt, so fängt er alle bekannten alten FPU–Befehle ab und sendet ein TRAP. Der TRAP–handler wird nun so umgeschrieben, daß er diese TRAPs erkennt und darauf reagiert, indem der entsprechende Befehl emuliert wird.

Anschließend wurden die oben beschriebenen Alternativen des Coprozessoranschlusses diskutiert. Als Ergebnis kam heraus, daß versucht werden sollte, beide Coprozessoren (FPU und SFU) zu betreiben. Die Möglichkeit die FPU auszubauen soll wirklich nur als letzte Möglichkeit in Betracht gezogen werden. Mit der genauen Abfolge und Syncronisation der Coprozessoren gabe es nich Unstimmigkeiten, die bis zum nächsten Treffen geklärt werden sollen. Die Gruppe Interface hat ein unbenutztes Bit gefunden. Dazu müssen aber noch weitere Tests durchgeführt werden. Da sich die Probleme bei den Gruppen Betriebssytem und Interface zum großen Teile überlagern, wurde beschlossen, daß die Gruppen enger zusammen arbeiten. Als Aufgabe für die nächste Woche wurden alle Gruppen beauftragt einen Entwurf für ein P30 Papier anzufertigen. Dazu wurden Hilfszettel, die den genauen Aufbau beschreiben an jede Gruppen ausgeteilt. Es sind auch noch weiter Bücher gefunden worden, die jetzt zum Teil bestellt werden sollen.
Ende: gegen 14:45