

STE-Speed - 1040STE auf Trab gebracht

Turmbauten verursachen häufig Probleme - eine Erfahrung, die so mancher Atari-Besitzer mit den ehemaligen Bewohnern Babylons teilt. Die Problematik ist zwar etwas anders gelagert, für den Betroffenen jedoch sehr leidig, denn oft verhindern ungünstige Gegebenheiten die Aufstockung des Prozessorsockels um die begehrte Beschleunigerkarte.

Besitzer eines 1040STE wissen davon ein Lied zu singen. Eine Beschleunigung des Rechners endet in der Regel mit dem Umzug in ein anderes Gehäuse, denn Beschleunigerkarten sind normalerweise für Prozessoren im DIL-Gehäuse ausgelegt. Zusammen mit dem dann notwendigen Adapter von PLCC auf DIL ist schnell eine Bauhöhe erreicht, die eine weitere Verwendung des Originalgehäuses unmöglich macht. Die hier vorgestellte Bastellösung ermöglicht den Einsatz eines mit 16MHz getakteten 68000ers im original Prozessorsockel des 1040STE. Die Systemleistung steigt dadurch um ca. 30%. Auf den ersten Blick sieht das recht mager aus, doch die Geschwindigkeitssteigerung ist deutlich spürbar. Außerdem bietet diese Bastelei weitere Vorteile: die Kosten belaufen sich auf ca. 60,- DM, es gibt keine Platzprobleme im Rechner und so nebenbei fällt noch ein HD-Modul ab.

Vorgeschichte

Die Idee für diese einfache Art der Beschleunigung kam mir bei dem Versuch einen ebenso einfachen Beschleuniger, den Speedup-16, in einen 1040 STE einzubauen. Bei dem Speedup-16 handelt es sich eigentlich um einen LowCost-Beschleuniger für den ST, also für einen Prozessor im DIL-Gehäuse, der auch bestimmte Eigenarten der ST-Hardware berücksichtigt. Der 1040 STE hingegen hat zum großen Teil eine modernere Hardware, die auch im Mega STE eingesetzt wird - und der läuft bekanntlich auch mit 16MHz! Was liegt also näher, als den Grundgedanken des Bastelprojektes aus der c't 10/90 neu aufleben zu lassen und den 1040 STE mit etwas mehr Dampf zu versorgen.

Wieviel bringt's ?

Im Schnitt ca. 30% Leistungssteigerung. Ist natürlich abhängig davon, was man mit dem Rechner anstellt. Wer beispielsweise seine Zeit mit dem Kopieren von Disketten totschrägt, hat nichts davon. Der Zugriff auf den Arbeitsspeicher ist ebenfalls von der Beschleunigung ausgenommen, da der Videoshifter durch seine Zugriffe hier das Tempo diktiert. Trotzdem bleibt noch genug zum Beschleunigen übrig. Die Benchmarks zeigen deutlich, wo der Gewinn liegt (Bild 1 und Bild 2). Wie man sieht ist der Leistungsgewinn ungleichmäßig verteilt. Die Entscheidung ob sich der Umbau lohnt wird möglicherweise noch dadurch beeinflusst, dass, quasi als Nebenprodukt, ein HD-Modul abfällt. Dieses HD-Modul ist zwar spartanisch ausgestattet (nur für ein Laufwerk, kein Autostep), ist aber in Verbindung mit TOS 2.06 ohne Software zum Umschalten der Steprate benutzbar, doch mehr dazu später.

Wie funktioniert's ?

Eigentlich bestechend einfach: Der Prozessor wird, statt nur mit 8MHz, wahlweise mit 8MHz oder 16MHz getaktet. Da der Prozessor im 16MHz-Betrieb die Gültigkeit der Daten deutlich früher als im

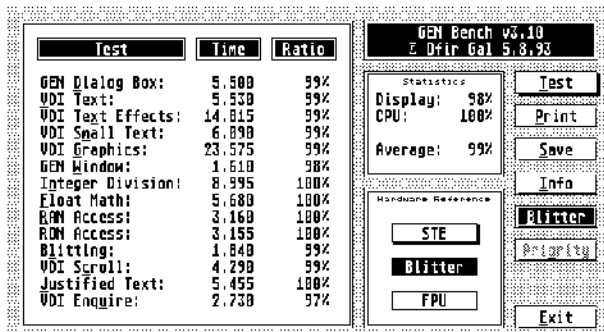


Bild 1: 1040 STE mit 8MHz

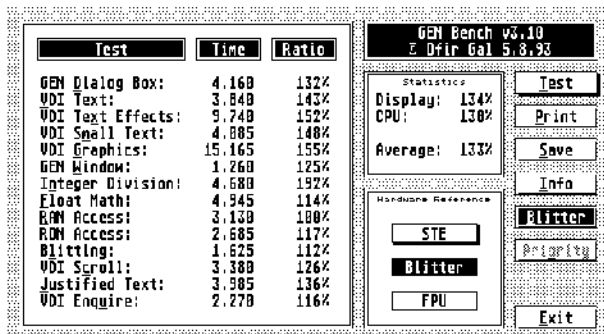


Bild 2: 1040 STE auf 16MHz beschleunigt

8MHz-Betrieb durch Aktivierung von Adress Strobe (/AS) signalisiert, reagieren die anderen Chips im STE auch viel früher durch Aktivierung von /DTACK (Daten gültig). Im 16MHz-Betrieb ist das glattweg gelogen, die Daten sind nämlich noch gar nicht gültig. Vielmehr sind die Bustreiber noch damit beschäftigt die Daten an ihren Ausgängen zu stabilisieren, so dass der Prozessor nur Müll zu lesen bekommt. Das Ergebnis ist ein besonders hartnäckiger Bildschirmschoner - der Monitor bleibt dunkel.

Um das Problem in den Griff zu bekommen, muss man den Prozessor also so lange "hinhalten", bis die Bustreiber ihre Arbeit erledigt haben und die gewünschten Daten tatsächlich auf dem Bus liegen. Dazu muss /DTACK so lange verzögert werden, bis dieser Fall eingetreten ist. Im 8MHz-Betrieb soll dagegen alles ohne Verzögerung ablaufen. Im Prinzip braucht man demnach nur einen Multiplexer, der einerseits die Taktfrequenz für den Prozessor auf die gewünschte Rate setzt, und außerdem im 8MHz-Betrieb /DTACK möglichst ohne Verzögerung und im 16MHz-Betrieb um ca. 125ns verzögert an den Prozessor weiter reicht.

Zur Hardware

Gemäß dieser Anforderung an die "Schaltung" lässt sich das Vorhaben sicherlich auch mit einem einfachen Multiplexer durchführen. In Bild 3 ist zu sehen, wie die Sache mit einem 74F157 funktionieren könnte. "Könnte" deshalb, weil ich es nicht ausprobiert habe.

Ich habe mich für den Einsatz eines Gals entschieden, weil sich damit besser experimentieren lässt, man muss es ggf. nur neu programmieren und spart dadurch 'ne Menge Verdrahtung. Nicht benutzte Pins eines Gals lassen sich darüber hinaus zweckentfremden, wie zum Beispiel in diesem Fall als HD-Modul, oder dienen der einfachen Realisierung von Verbesserungen. Für unseren Zweck reicht ein Gal 16V8 mit einer Zugriffszeit von 25ns. Zum besseren Verständnis hier die Erläuterung der Funktion der einzelnen Pins:

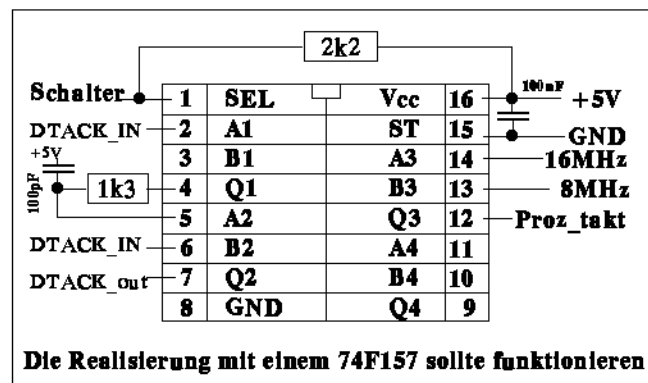


Bild 3: Probieren geht über Studieren

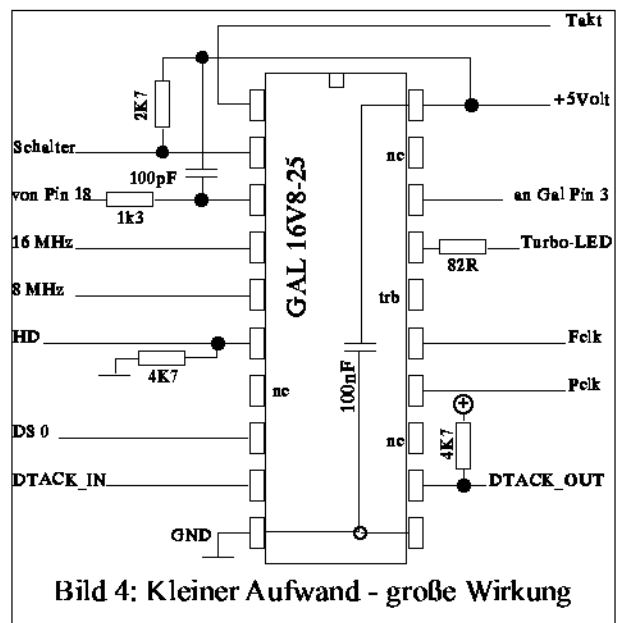
- Pin 1 (clk): Hat eigentlich nur die Funktion ein Umschalten des Prozessortaktes zu ermöglichen, ohne dass der Rechner abschmiert. Funktioniert (fast) immer. Clk sollte möglichst niedrig und phasenrichtig zum Systemtakt sein. Ich habe dafür die 500kHz von Pin 6 der GSTMCU hergenommen. Wer nicht an dem 144poligen Ungetüm herumlöten möchte, sollte wie folgt verfahren: Ein Stück von dem IC weg mit einem Schraubendreher den Lötstoplack von der entsprechenden Leiterbahn abkratzen und die Leitung direkt auf die Leiterbahn löten.
- Pin 2 (tbo): Hier wird der Turbo-Schalter angeschlossen. Bei offenem Kontakt liegt der Pin über den Pullup-Widerstand auf 5 Volt und der Prozessor wird mit 8MHz getaktet. Wird über den Schalter Masse angelegt, liegen 16MHz am Prozessor an.
- Pin 3 (v_dtack): Das über das RC-Glied verzögerte dtack wird hier eingespeist und wieder in eine vernünftige Rechteckform gebracht. Dieser Pin ist über 1k3 mit Pin 18 verbunden.
- Pin 4 (clk16): Eine Verbindung mit Pin 2 der GSTMCU sorgt dafür, daß hier 16MHz anliegen.
- Pin 5 (clk8): Hier landen 8MHz, entweder von Pin 3 der GSTMCU oder von Pin 34 des Blitter-Sockels (muss nach dem Auftrennen der Leiterbahnen an Pin 39 des DMA-Chips weitergereicht werden!).
- Pin 6 (hd): Der Pin 2 des Shugart-Bus liefert für diesen Pin das hd-Signal, soweit ein geeignetes HD-Laufwerk angeschlossen ist.
- Pin 7 ist nicht belegt.
- Pin 8 (ds0): Drive Select 0 findet sich an Pin 10 des Shugart-Bus oder an Pin 20 des Soundchip. Wird bei Zugriff auf Laufwerk A: aktiv.

- Pin 9 (dtack): Mit diesem Signal wird dem Prozessor die Gültigkeit der Daten mitgeteilt. Dtack findet sich an Pin 27 des Blitter-Sockels (muss nach dem Auftrennen der Leitbahnen an Pin 46 des MFP weitergereicht werden!).
- Pin 10 und 11 sind Masse.
- Pin 12 (o_dtack): Dies ist das neue DTACK für den Prozessor und wird mit Pin 10 des Prozessors verbunden. Die Durchkontaktierung b in Bild 6 führt direkt dort hin.
- Pin 13 ist nicht belegt.
- Pin 14 (pclk_out): Der schaltbare Prozessortakt gehört an Pin 15 desselben, zu erreichen über Durchkontaktierung a in Bild 6.
- Pin 15 (fclk_out): Geht an Pin 18 (Takt) des Floppycontrollers. Der Pin 18 darf nicht mehr mit der Hauptplatine verbunden sein, also durchkneifen oder aus dem Sockel hängen lassen.
- Pin 16 (trb): Dient nur der Synchronisation des Umschaltvorganges und um Pellen des Schalters zu minimieren. Wird nix angeschlossen.
- Pin 17 (led): Das Wichtigste an der Schaltung: Die Turbo-LED. Wird hier mit 8MHz getaktet, um aus dem Gal genügend Strom für ein Minimum an Leuchtkraft herauszukitzeln.
- Pin 18 (z_dtack): Folgt dem normalen dtack und wird dann über das RC-Glied an Pin 3 des Gals weitergereicht.
- Pin 19 ist nicht belegt.
- Pin 20: + 5 Volt.

In Bild 4 ist die Außenbeschaltung des Gals dargestellt. Fünf Widerstände und zwei Kondensatoren machen das Ätzen einer Platine überflüssig. Entsprechend dem Schaltungsaufwand gestaltet sich auch die Stückliste recht übersichtlich: Beim Aufbau der Schaltung sollte man die Bestückung auf der Lötseite der Platine vornehmen. Die Pins des IC-Sockels werden soweit gekürzt, dass sie nicht mehr durch die Platine ragen. Den 100nF-Kondensator kann man bereits jetzt zwischen Pin 10 und 20 des IC-Sockels anlöten, dann wird der Sockel auf die Platine gelötet (bei Platinen mit Lötstreifenraster nicht vergessen, die Leitbahnen unter dem Sockel zu durchtrennen!). Den Rest der Bestückung kann man am besten nach dem Einbau erledigen.

Vorbereitung des Rechners

Fast schon eine liebgewonnene Gewohnheit für Atari-Besitzer: Das Durchtrennen von Leiterbahnen. Auch bei diesem Projekt wollen wir darauf nicht verzichten. Leider hängt der Prozessor des 1040 STE mit seiner Taktversorgung und mit dem Dtack-Signal in der Mitte einer Kette von Bauelementen. Das bedeutet, dass die Leiterbahnen für Takt und Dtack vor und hinter dem Prozessorsockel durchtrennt werden müssen. Außerdem müssen das Takt- und das Dtack-Signal



- 1 Stück Platine Lötstreifenraster
- 1 68HC000-16
- 1 Gal 16V8-25
- 1 IC-Sockel 20pol
- 1 Widerstand 1K3
- 1 Widerstand 2K7 (1K - 4K7)
- 1 Widerstand 82R
- 2 Widerstand 4K7
- 1 Kondensator 100pF
- 1 Kondensator 100nF
- 1 Schalter ein/aus
- dünnes Kabel/Draht

sozusagen an dem Prozessor vorbei an die hinter dem Prozessor liegenden Bauelemente weitergereicht werden. In Bild 6 (am rechten Rand, Bild 5 ist die Stückliste) findet sich ein Vorschlag, wo die Leiterbahnen aufzutrennen sind, in jedem Fall aber selbst nachmessen, ich weiß nicht, wie viele Platinenlayouts es gibt.

Nach dem Durchtrennen der Leiterbahnen für Takt und Dtack sollten Pin 15 und Pin 10 des Prozessors mit keinem anderen Bauelement mehr verbunden sein. Leider bekommt der DMA-Chip jetzt auch kein Taktsignal mehr und der 68901 (MFP) wird sein Dtack nicht mehr los. Abhilfe für dieses Problem schaffen zwei Drahtbrücken; Pin 39 des DMA-Chips muss mit Pin 34 des Blitter-Sockels oder mit Durchkontaktierung c in Bild 6 verbunden werden und Pin 46 des MFP bekommt eine Verbindung zu Pin 27 des Blitter-Sockels. Damit ist die "harte Variante" der Rechnervorbereitung abgeschlossen. Für Softies gibt es auch noch eine andere Möglichkeit. Dabei

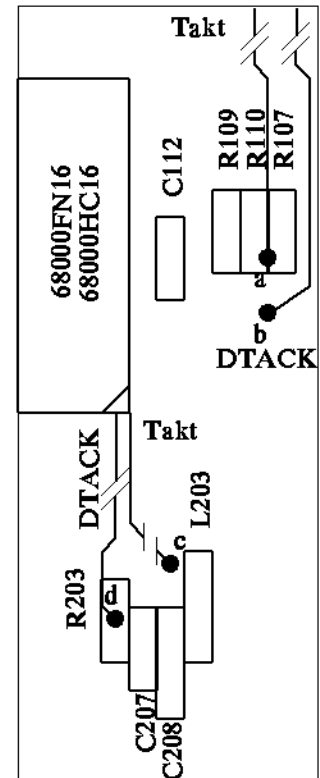
wird an Pin 10 und Pin 15 des 68000-16 je ein dünner Draht angelötet. Diese Pins werden durch schmale Plastikstreifen oder sonstwie gegen die Federkontakte des Prozessorsockels isoliert, Prozessor in den Sockel drücken, fertig.

Einbau der Schaltung

Links vom Prozessor sollte eigentlich genügend Platz sein. Das Platinchen also mit Heißkleber oder Kaugummi irgendwo festpappen, den Rest der Bestückung durchführen und verdrahten. Spannungsversorgung nicht vergessen und darauf achten, daß das Gal noch in den Sockel gesteckt werden kann. Ein erster Probelauf ohne Gal sollte auf Anhieb funktionieren. Dazu folgende Pins des Gal-Sockels miteinander per Drahtbrücke verbinden: Pin 5 (clk8) mit Pin 14 (pclk_out) und Pin 9 (dtack_in) mit Pin 12 (dtack_out). Wenn bisher kein grober Fehler passiert ist, läuft der Computer wie vor dem Eingriff. Wenn nicht, ist mit großer Wahrscheinlichkeit eine falsche Leiterbahn durchtrennt worden oder beim Weiterreichen des Takt- oder Dtacksignals ist etwas schief gegangen, also noch einmal kontrollieren. Falls Kaugummi als Haftmedium eingesetzt wird, ist ein übermäßiges Einbringen von Speichel in den Rechner zu vermeiden. Speziell bei Kontakt mit Bauelementeanschlüssen an denen statische Potentiale anliegen, verursachen die im Speichel enthaltenen Elektrolyte Migrationseffekte, die zu gefährlicher Instabilität des Systems führen können. Als besonders kritisch, weil stark speichelfördernd, ist hier "Hubba Bubba" zu klassifizieren!

Das Gal

Die Logikgleichungen beinhalten keine besonderen Tricks, der Hilfsausgang "trb" ist nötig, damit der Rechner beim Umschalten der Taktfrequenz nicht sofort stehen bleibt. Eleganter wäre eigentlich die Lösung die 8MHz aus den 16MHz im Gal zu erzeugen, aber in 50% aller Fälle ist die Phasenlage des 8MHz-Signals nach dem Einschalten dann genau verkehrt herum. Den Floppycontroller stört das nicht weiter, der Rest des Rechners quittiert einen solchen Umstand jedoch mit Arbeitsverweigerung.



```

%ID S_P_V1.1
%TYP gal16v8
%PINS
clk: 1
tbo: 2
!v_dtack: 3
clk16: 4
clk8: 5
hd: 6
!ds0: 8
!dtack: 9
!o_dtack: 12
pclk_out: 14
fclk_out: 15
trb: 16
led: 17
!z_dtack: 18

%LOGIC
trb <- tbo;
pclk_out = trb * clk8
    + !trb * clk16;
o_dtack.OE = GND;
o_dtack = trb * dtack
    + !trb * v_dtack;
z_dtack = dtack;
fclk_out = hd * ds0 * clk16
    + !hd * ds0 * clk8
    + !hd * !ds0 * clk8
    + hd * !ds0 * clk8;
led = !trb * clk8;
%END

```

*/ Bild 7: S_P_VI_1.LCI Bilduntertitel: "Bild 7: Logikgleichung für Maxon Gal-Prommer"

Das HD-Modul

Der Ausdruck wirkt ein wenig dick aufgetragen, schließlich handelt es sich dabei gerade einmal um drei Pins unseres Gals, aber letztlich zählt das Ergebnis. Um nun tatsächlich HD-Disketten lesen und beschreiben zu können, sind noch einige Bedingungen zu erfüllen. Als erstes muss natürlich ein HD-Floppylaufwerk her. Dieses Laufwerk muß an Pin 2 des Shugart-Bus den Typ der eingelegten Diskette signalisieren. Bei einer ganzen Reihe der heute erhältlichen Floppies ist Pin 2 leider nicht belegt und man muß das HD-OUT-Signal über eine Drahtbrücke nachreichen. Im Zweifelsfall also den kompetenten Atari-Fachhändler um die Ecke zu Rate ziehen. Zweitens sollte der Rechner über einen HD-fähigen Floppycontroller verfügen. Da gibt es nur einen Typ, der 100%ig funktioniert, Ajax genannt. Der WD 1772-02-02 funktioniert in einigen Fällen auch im HD-Betrieb, aber beileibe nicht so oft und so sicher, wie allgemein angenommen wird. In jedem Fall muß Pin 18 des Floppycontrollers mit Pin 15 (felk_out) des Gal verbunden werden, dabei darf der Pin keinen Kontakt mehr zum Motherboard haben (durchkneifen oder aus dem Sockel hängen lassen). Wenn im Rechner noch TOS 1.06 eingebaut ist, ist es erforderlich die Steprate per Software von 3ms auf 6ms heraufzusetzen, ist TOS 2.06 eingebaut, kann man sich auch das sparen, wenn die Lötbrücke E6 in der Nähe der Eproms geschlossen wird. Das TOS 2.06 bietet dann im Formatierdialog auch den Menüpunkt "Hohe Schreibdichte" an.

Einige Anmerkungen

Es ist mir zuweilen untergekommen, daß der Rechner bombt, wenn er im 16MHz-Betrieb eingeschaltet wird. Die Ursache habe ich nicht gefunden, doch ein Druck auf den Reset-Knopf läßt ihn dann einwandfrei hochlaufen. Die Schaltung ist natürlich nicht in allen möglichen und unmöglichen Hard- und Softwarekonfigurationen getestet worden, möglicherweise ist es schlau vor dem Umbau ein Backup der Festplatte zu machen.

Die JEDEC-Datei zur Programmierung des Gals und eine Kurzfassung der Einbauanleitung sind in der ST-Computer Mailbox zu finden.

Literatur: M68000 User's Manual, Motorola, Inc., c't 10/90 S.330 ff, Modellpflege, O. Winter, R. Kratzer

us