

# Einführung in Computer Microsystems

## 6. Aufgabenblatt

### Sommersemester 2007

#### Aufgabe: Discount

Der aus der Vorlesung (Foliensatz 5 bis Folie 40) bekannte Video-Controller „Discount“ soll nun variable Auflösungen ausgeben können. Da diese je nach Anwendung und angeschlossenem Monitor auch zur Laufzeit wechseln können, muss Discount um eine Programmierschnittstelle z.B. für Mikroprozessoren erweitert werden. Gehen Sie dazu wie folgt vor:

- Öffnen Sie das auf der Homepage des FG ESA bereitgestellte ISE-Projekt `discount.zip`, welches die Verilog-Quellen von Discount enthält. In der Datei `discount_defs.v` beschreiben die Konstanten ``H1`, ``H`, ``H2`, ``H3` und ``Hline` den horizontalen Bildaufbau, die Konstanten ``V1`, ``V`, ``V2`, ``V3` und ``Vframe` den vertikalen Bildaufbau. ``Msz` beschreibt die Größe des Bildspeichers. Ersetzen Sie diese Konstanten in den Modulen `discount`, `hcount`, `vcount` und `memacc` durch Register der Bitbreite 10 (Breite ``Asz` im Fall von Register `Msz`). Durch einen Reset via `NRST` sollen alle neuen Register den Wert der jeweiligen Konstanten als Default erhalten.
- Erweitern Sie die oben genannten Module um einen Programmierbus, bestehend aus den Unterbussen `PADDR` und `PDATA` sowie dem Signal `WE`, um die neuen Register beschreiben und wieder auslesen zu können (Modul `memacc` nur beschreiben). Das Signal `WE` gibt an, ob ein Register beschrieben (`WE == 1`) oder gelesen (`WE == 0`) werden soll. Der Adressbus `PADDR` überträgt die Adressen der einzelnen Register, ``H1`, ``H`, ``H2` und ``H3` erhalten die Adressen 0 bis 3 und ``V1`, ``V`, ``V2` und ``V3` die Adressen 4 bis 7. Der Datenbus `PDATA` ist bidirektional auszulegen, überträgt also sowohl Schreib- als auch Lesedaten. Entwickeln Sie für jedes der angesprochenen Module eine Adressdekodierlogik (siehe dazu auch Foliensatz 3, Abschnitt „Busse“ ab Folie 24), die bei Lesezugriffen im folgenden Takt auf `PDATA` den Wert des

gewünschten Registers via Tri-State-Treiber bereitstellt und bei Schreibzugriffen im selben Takt den Wert auf **PDATA** in das adressierte Register speichert.

- Die neuen Register **msz**, **hline** und **vframe** werden nicht direkt beschrieben, ihre Werte sollen laufend in der Hardware aus den übrigen neuen Registern berechnet werden, analog zur Berechnung der Konstanten in **discount\_defs.v**.
- Passen Sie die Testbench an, indem Sie den Programmierbus hinzufügen. Testen Sie zunächst mit *inaktivem* Programmierbus, ob die Default-Auflösung 640\*192 (Reset-Werte der neuen Register) noch korrekt ausgegeben wird. Schauen Sie sich auch die erzeugte PBM-Datei an.
- Schreiben Sie nun einen neuen **initial**-Block für die Testbench, der eine Auflösung von 320\*200 Punkten über den Programmierbus einstellt (größere Auflösungen sind theoretisch mit breiteren Registern möglich, leider wird der ISE-Simulator dann sehr langsam und unbenutzbar). Verwenden Sie dazu folgende Werte für die einzelnen Register:

| H1 | H   | H2 | H3 | V1 | V   | V2 | V3 |
|----|-----|----|----|----|-----|----|----|
| 48 | 320 | 44 | 44 | 60 | 200 | 50 | 16 |

Lesen Sie die Register **h** und **v** über den Programmierbus wieder zurück und geben Sie die hoffentlich korrekten Werte mit **\$display** aus. Vergleichen Sie die nun erzeugte PBM-Datei.