

# Grundlagenpraktikum Rechnerarchitektur (GRA)

## Tutorübung

Moritz Beckel

13.06.2022 16:00 / 10.06.2022 15:00

# Hausaufgaben

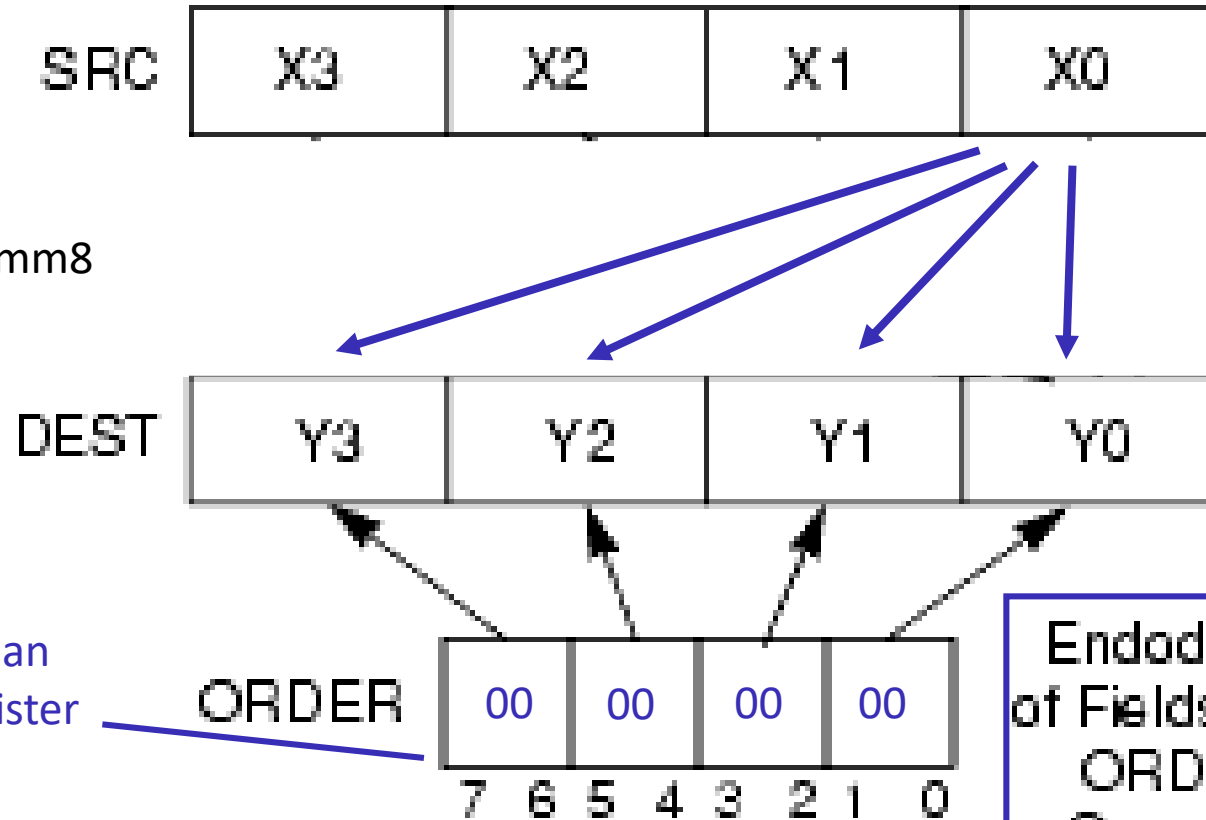
- Toupper (SSE)
- Strlen (SIMD)
- Quiz

# Hausaufgaben

PSHUFD

xmm1, xmm0, imm8

Position im  
immediate gibt an  
welches Zielregister  
angesprochen  
werden soll



Jedes Quellregister  
ist mit 2 Bit kodiert

Encoding	00B - X0
of Fields in	01B - X1
ORDER	10B - X2
Operand	11B - X3

# T7.1 Valgrind

**Material:** <https://gra.caps.in.tum.de/m/outerprod.tar>

1. Besprechen Sie mit Ihrem Tutor zunächst den Aufbau und die grobe Funktionsweise des Programms.
2. Kompilieren Sie das Programm mit `make` und führen Sie es aus. Verhält es sich wie erwartet?
3. Führen Sie das Programm zum Debuggen nun mit GDB aus (`gdb ./outerprod; r`). An welcher Stelle der Datei `outerprod.c` stürzt das Programm ab?
4. Lassen Sie sich mit dem Befehl `p mat` den Pointer auf die Ausgabematrix anzeigen. Können Sie allein hieraus erkennen, warum das Programm abstürzt?

## T7.1 Valgrind

5. Führen Sie das Programm nun mit Valgrind aus: `valgrind ./outerprod`. Wie interpretieren Sie die Ausgabe? Was ist die *unmittelbare* Ursache für den Absturz?
6. Wie groß ist der Speicherbereich, auf den zum Zeitpunkt des Programmabsturz zugegriffen wird? Wie groß *sollte* er sein?
7. Wie erklären Sie sich diese Diskrepanz?
8. Informieren Sie sich im GCC Manual<sup>2</sup> über die Integer Overflow Built-ins und überlegen Sie, wie Sie diese nutzen können, um dieses und ähnliche Probleme allgemein zu verhindern.

# Hausaufgaben

## P8.1 Hamming-Distanz [4 Pkt.]

**Aufgabe:** Optimieren Sie folgende Funktion, welche die Anzahl der unterschiedlichen Elemente der Arrays a und b der gleichen Länge n berechnet, um die Überlebensfähigkeit der ERA-Pinguine zu erhöhen!

```
1 size_t hamming_dist(size_t n, const char a[n], const char b[n]) {  
2     size_t res = 0;  
3     for (size_t i = 0; i < n; i++)  
4         res += a[i] != b[i];  
5     return res;  
6 }
```

# Hausaufgaben

## **X8.1 ToUpper: MemeAssembly-Edition [2 Pkt. Bonus]**

In dieser Einheit betrachten wir erneut die exzellente Funktion `toupper`, welche in einem String sämtliche unexzellente Klein-Buchstaben durch die entsprechenden exzellenten Groß-Buchstaben ersetzt. Diese Aufgabe wird Ihnen eine Einführung in die exzellente Programmiersprache MemeAssembly geben.