

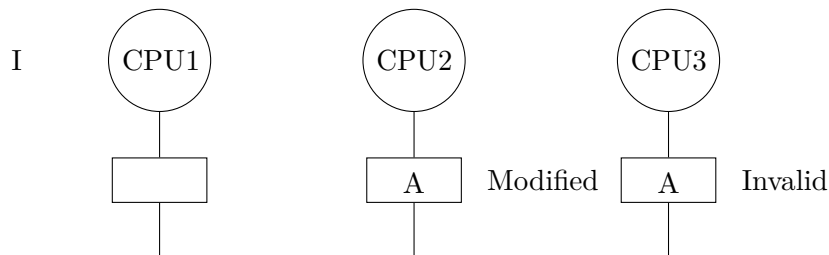
Aufgabe 1: MESI-Protokoll

Beantworten Sie folgende Fragen, geben Sie ggf. Ihre Quellen an.

- Was ist unter Cache Konsistenz zu verstehen?
- Was ist unter Cache Kohärenz zu verstehen?
- Beschreiben Sie kurz *in eigenen Worten* die möglichen Zustände im MESI-Protokoll.
- Warum darf es im MESI-Protokoll den Zustand *Shared Modified* nicht geben?

Nehmen Sie an, die Prozessoren CPU1, CPU2 und CPU3 greifen auf einen gemeinsamen Speicher zu. Außerdem hat jede CPU einen eigenen lokalen Cache.

Wie in der Grafik ersichtlich liegt das Datum A im lokalen Cache von CPU2 und CPU3 vor. Nehmen Sie an, auf dieses Datum bezogen ist CPU2 im Zustand *Modified* und CPU3 im Zustand *Invalid*.



Zeigen Sie grafisch und erläutern Sie kurz, wie sich die Zustände aufgrund der folgenden Befehlsfolge verändern:

- CPU3 schreibt Datum A zu A'
- CPU1 liest A'
- CPU1 schreibt A' zu A''

Sie finden einen \LaTeX -Snippet für diese TikZ-Grafik im Anhang des Zettels.

Aufgabe 2: Assembler-Code lesen

Gegeben sei folgende NASM-Funktion (Parameter stehen in rdi und rsi, der Rückgabewert steht in rax):

```
1          global function
2
3function:  mov rax, rsi
4          add rax, rdi
5          sub rax, 1
6
7.loop:    cmp rax, rsi
8          jbe .end
9
10         mov r8b, [rsi]
11         mov r9b, [rax]
12         mov [rax], r8b
13         mov [rsi], r9b
14
15         inc rsi
16         dec rax
17         jmp .loop
18
19.end:     ret
```

Die Funktion wird folgendermaßen aufgerufen:

```
uint8_t array[] = {1,2,3,4,5,6,7,8};  
function(8, array);
```

Beantworten Sie folgende Fragen:

1. Geben Sie den Inhalt des Arrays nach dem ersten Schleifendurchlauf von `.loop` an.
2. Welchen Rückgabewert gibt die Funktion zurück?
3. Wie sieht der Inhalt der Variable `array` nach dem Funktionsaufruf aus?
4. Optional: Was tut das Programm?