

Lehre

Komplexpraktikum

SpartanMC

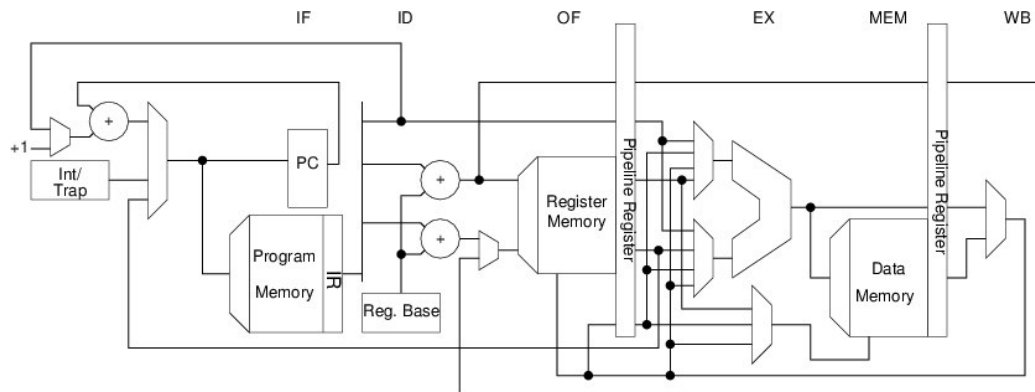
[Allgemeine Informationen](#)
[Aktuelles](#)
[Hardware](#)
[Software](#)
[Dokumente](#)
[Ideensammlung](#)
[Troubleshooting](#)
[Sprachdefinition](#)

Umstellung auf neue Pipeline



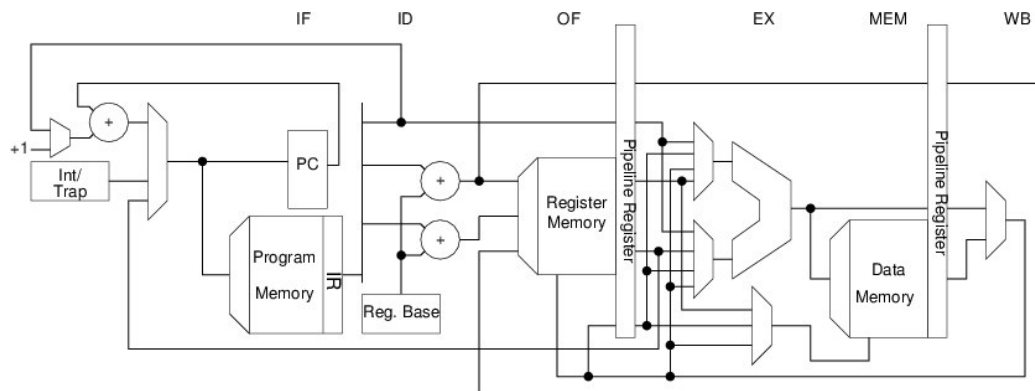
SpartanMC » Hardware » Umstellung auf neue Pipeline

SpartanMC Core mit 2 Port Registerfile



In dieser Konfiguration wird am Registerfile ein Phasensplitting durchgeführt, um in jedem Takt 2 Register zu lesen und ein Register zu schreiben. Dazu werden alle 4 Taktsignale **clk1**, **clk2**, **clk2x** und **phase** benötigt. Die Variante ist dadurch langsamer als die folgende Variante mit dem 3 Port Registerfile. Die 2 Port Variante benötigt aber einen Blockram weniger.

SpartanMC Core mit 3 Port Registerfile



In dieser Konfiguration arbeiten alle Funktionseinheiten des SpartanMC nur noch mit der positiven Flanke des Taktsignals **CLK1**. Bei dieser Konfiguration kann **CLK1** vom SpartanMC mit **CLKFX** eines DCM-Moduls verbunden werden. **CLK2**, **CLK2X** und **PHASE** des SpartanMC können unbeschaltet bleiben. In dieser Implementierung des SpartanMC mit 3 Port Register File stehen jetzt drei Signale zur Verfügung, mit denen der Systemtakt angehalten oder auf eine niedrigere Frequenz zur Laufzeit umgeschaltet werden kann. Damit ist es jetzt möglich die CPU anzuhalten, während eine spezielle IO-Komponente Daten im Speicher erzeugt. Es kann aber auch zur Laufzeit auf eine niedrigere Taktfrequenz umgeschaltet werden, um langsame Funktionseinheiten direkt ansprechen zu können. Die drei neuen Signale sind:

1. **step_en_off** Durch einen High-Impuls wird in den normalen Betriebsmode umgeschaltet.
2. **step_en_on** Durch einen High-Impuls wird in den STEP-Mode umgeschaltet.
3. **step** Mit jeder low-high Flanke an diesem Eingang wird ein Takt für den SpartanMC freigegeben.

Die drei Signale sollten mit **GND** verbunden werden, wenn sie nicht verwendet werden sollen.

Mit einem Schalter kann im jConfig eine der beiden Varianten für den SpartanMC Core ausgewählt werden. Achtung, die Beschaltung der Taktsignale wird nicht überwacht!

Aktuelle Informationen zur Implementierung

- [Geräte für new_pipe](#)
- [Konfigurationen aus dem SVN die auf new_pipe umgestellt sind](#)
- [Pipeline des neuen SpartanMC](#)
- [Dekodierung der IO-Adressen für die neue Pipeline](#)

Lehre

Komplexpraktikum

SpartanMC

Allgemeine Informationen

Aktuelles

Hardware

Software

Dokumente

Ideensammlung

Troubleshooting

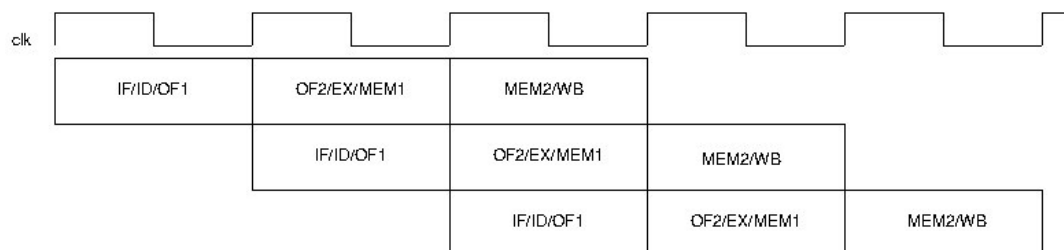
Sprachdefinition

Pipeline des SpartanMC



SpartanMC » Hardware » Prozessor » Pipeline des SpartanMC

Takte der Pipeline



* **IF** - [Instruktion Fetch](#)

* **ID** - [Instruktion Decode](#)

* **OF1** - [Operanden Fetch 1](#) (Konstanten und Bypass1 vorbereiten)

* **OF2** - [Operanden Fetch 2](#) (Auswahl Register, Bypass1, Bypass2 oder Konstante)

* **EX** - [Execute Instruction](#) (Ausführung der Befehle oder Berechnung von Speicheradressen)

* **MEM1** - [Memory 1](#) (Adressierung des Speichers (der I/O) und schreiben der Daten im Store-Befehl)

* **MEM2** - [Memory 2](#) (Lesen der Daten von der Adresse im Load-Befehl)

* **WB** - [Write Back](#) (Zurückschreiben der Ergebnisse in das Zielregister)

Lehre

Komplexpraktikum

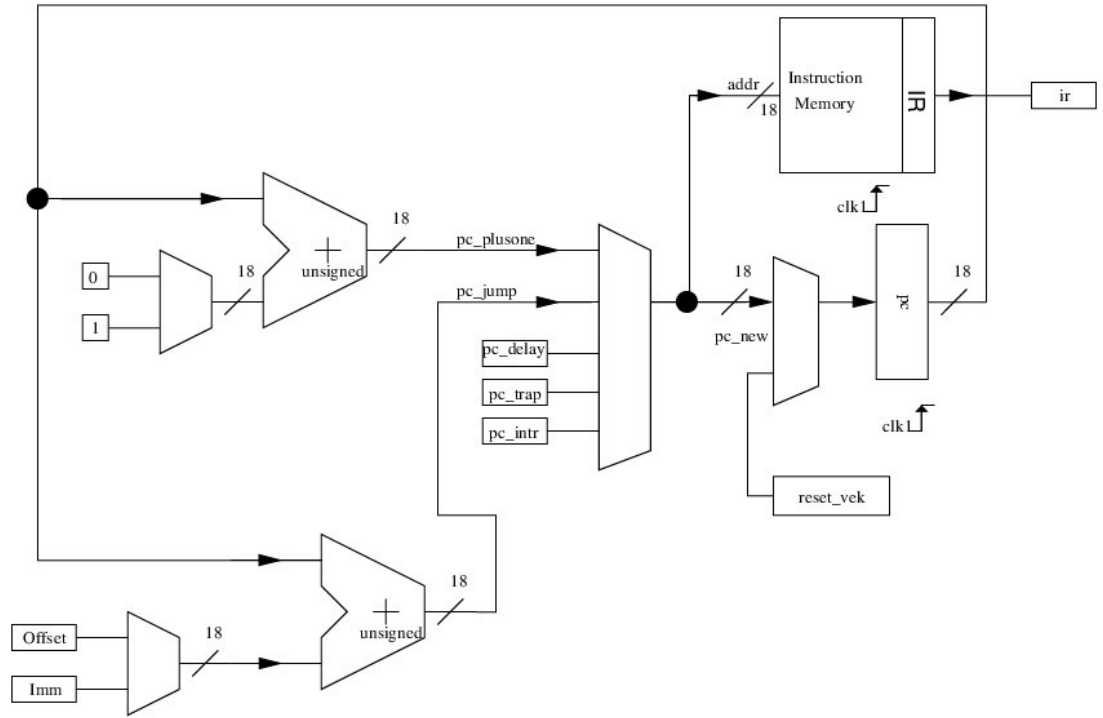
SpartanMC

- Allgemeine Informationen
- Aktuelles
- Hardware
- Software
- Dokumente
- Ideensammlung
- Troubleshooting
- Sprachdefinition

Pipeline des SpartanMC

SpartanMC » Hardware » Prozessor » Pipeline des SpartanMC

IF — Befehl lesen



Lehre

Komplexpraktikum

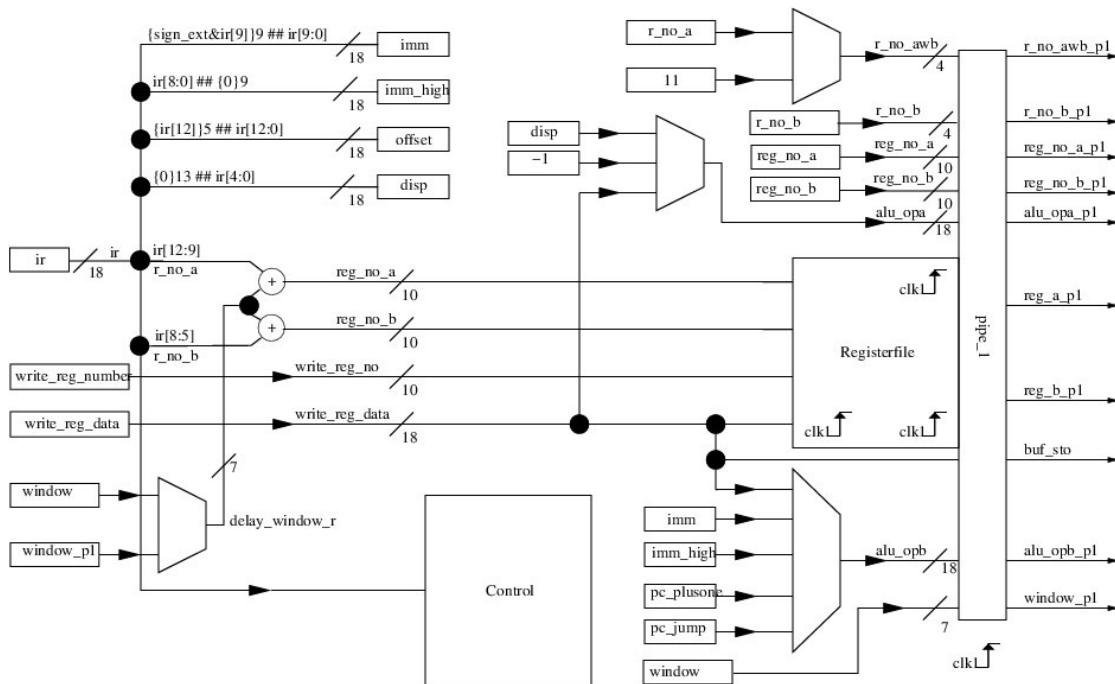
SpartanMC

- Allgemeine Informationen
- Aktuelles
- Hardware
- Software
- Dokumente
- Ideensammlung
- Troubleshooting
- Sprachdefinition

Pipeline des SpartanMC

SpartanMC » Hardware » Prozessor » Pipeline des SpartanMC

ID/OF1 — Befehl dekodieren/Konstanten oder Bypass 1 vorbereiten und Register adressieren



Lehre

Komplexpraktikum

SpartanMC

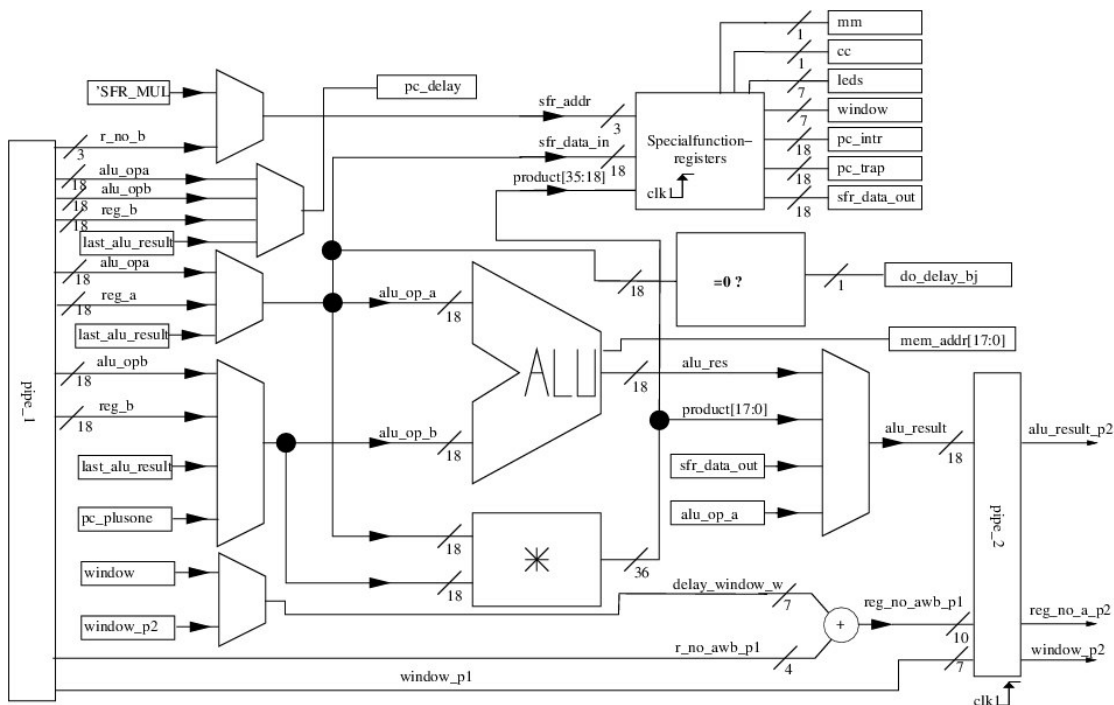
- Allgemeine Informationen
- Aktuelles
- Hardware
- Software
- Dokumente
- Ideensammlung
- Troubleshooting
- Sprachdefinition

Pipeline des SpartanMC



SpartanMC » Hardware » Prozessor » Pipeline des SpartanMC

OF2/EX — Register, Bypass1, Bypass2 oder Konstante auswählen/Befehl ausführen



Lehre

Komplexpraktikum

SpartanMC

Allgemeine Informationen

Aktuelles

Hardware

Software

Dokumente

Ideensammlung

Troubleshooting

Sprachdefinition

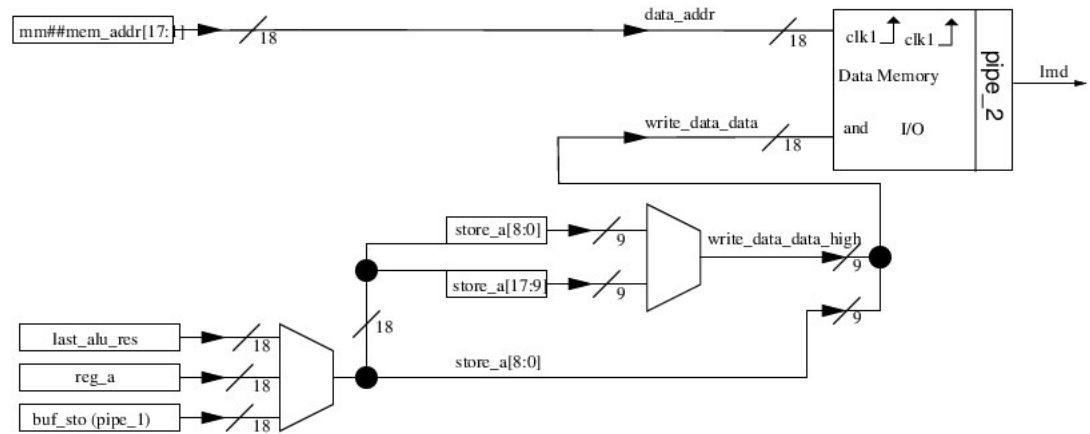
Pipeline des SpartanMC





 SpartanMC » Hardware » Prozessor » Pipeline des SpartanMC

MEM1 — Speicher adressieren und schreiben bei STORE



page: 5/6

Lehre

Komplexpraktikum

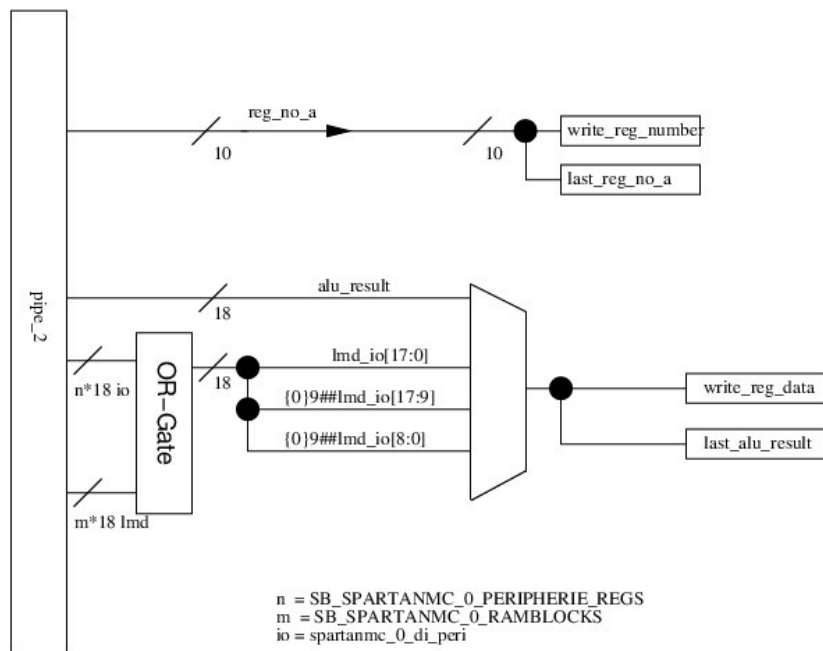
SpartanMC

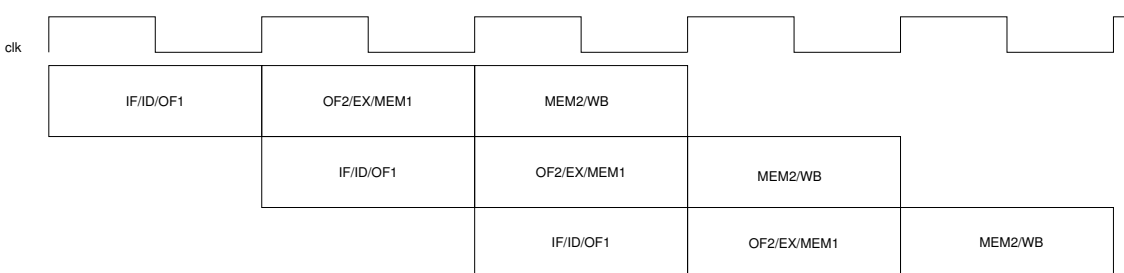
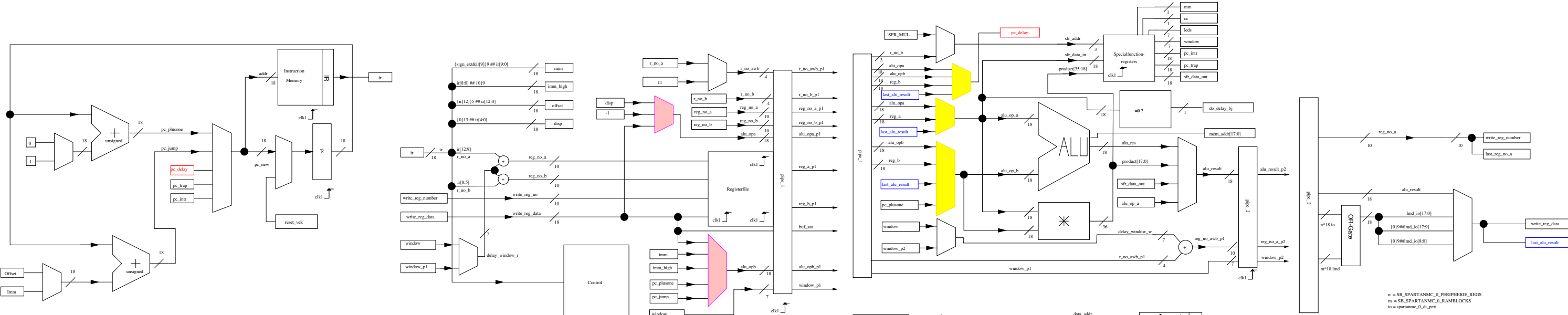
- Allgemeine Informationen
- Aktuelles
- Hardware
- Software
- Dokumente
- Ideensammlung
- Troubleshooting
- Sprachdefinition

Pipeline des SpartanMC

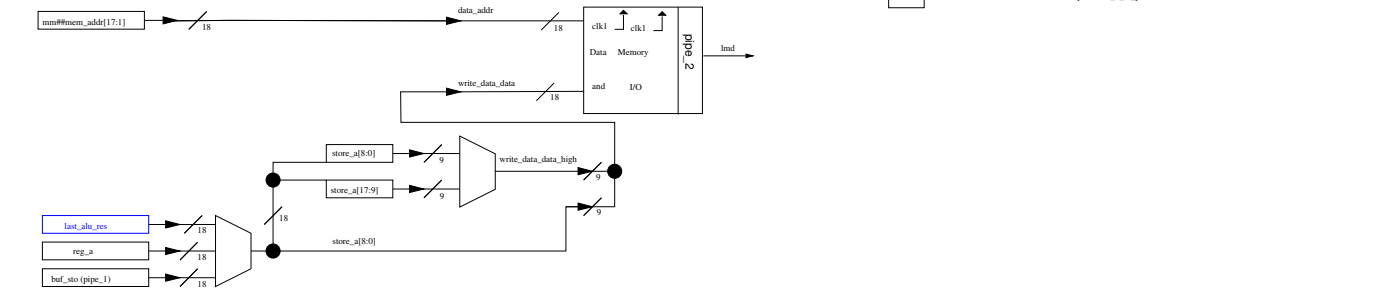
SpartanMC » Hardware » Prozessor » Pipeline des SpartanMC

MEM2/WB — Speicher lesen/Zurückschreiben der Ergebnisse





- **IF** - Instruction Fetch
- **ID** - Instruction Decode
- **OF1** - Operanden Fetch 1 (Konstanten und Bypass1 vorbereiten)
- **OF2** - Operanden Fetch 2 (Auswahl Register, Bypass1, Bypass2 oder Konstante)
- **EX** - Execute Instruction (Ausführung der Befehle oder Berechnung der Speicheradressen)
- **MEM1** - Memory 1 (Adressierung des Speichers (der IO) und schreiben der Daten im Store-Befehl)
- **MEM2** - Memory 2 (Lesen der Daten von der Adresse im Load-Befehl)
- **WB** - Write Back (Zurückschreiben der Ergebnisse in das Zielregister)



n = SB_SPARTANMC_0_PERIPHERE_REGS
 m = SB_SPARTANMC_0_RAMBLOCKS
 io = spartanmc_0_io_peri