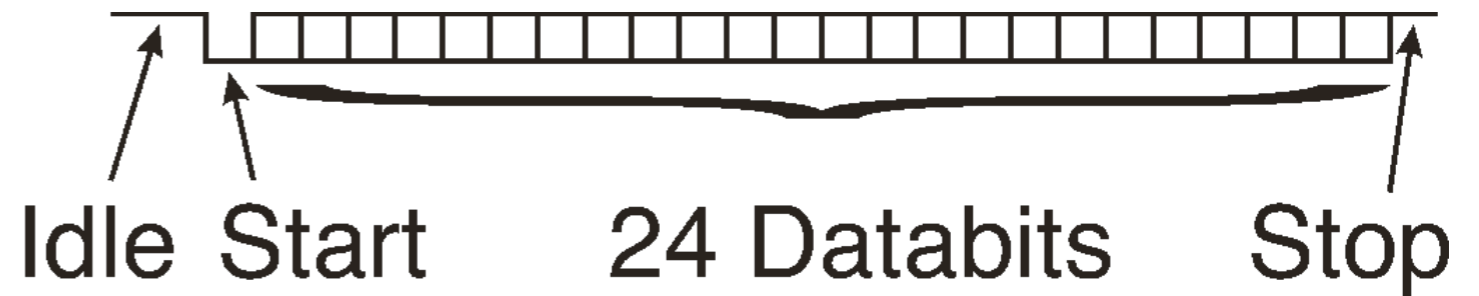


# **Ein intelligenter Auslese-ASIC für den GET4-TDC**

**Holger Flemming**

# Motivation

- Bisher:
  - Jeder GET4 benötigt eigenen seriellen Link zum ROC
  - Events werden als einzelne asynchrone Datenworte übertragen
  - Große Anzahl ROCs erforderlich



- Ziel:
  - Einsparung von ROCs
  - Daisy Chain oder Token Ring-Betrieb
- Dafür notwendig:
  - Bildung von Datenrahmen

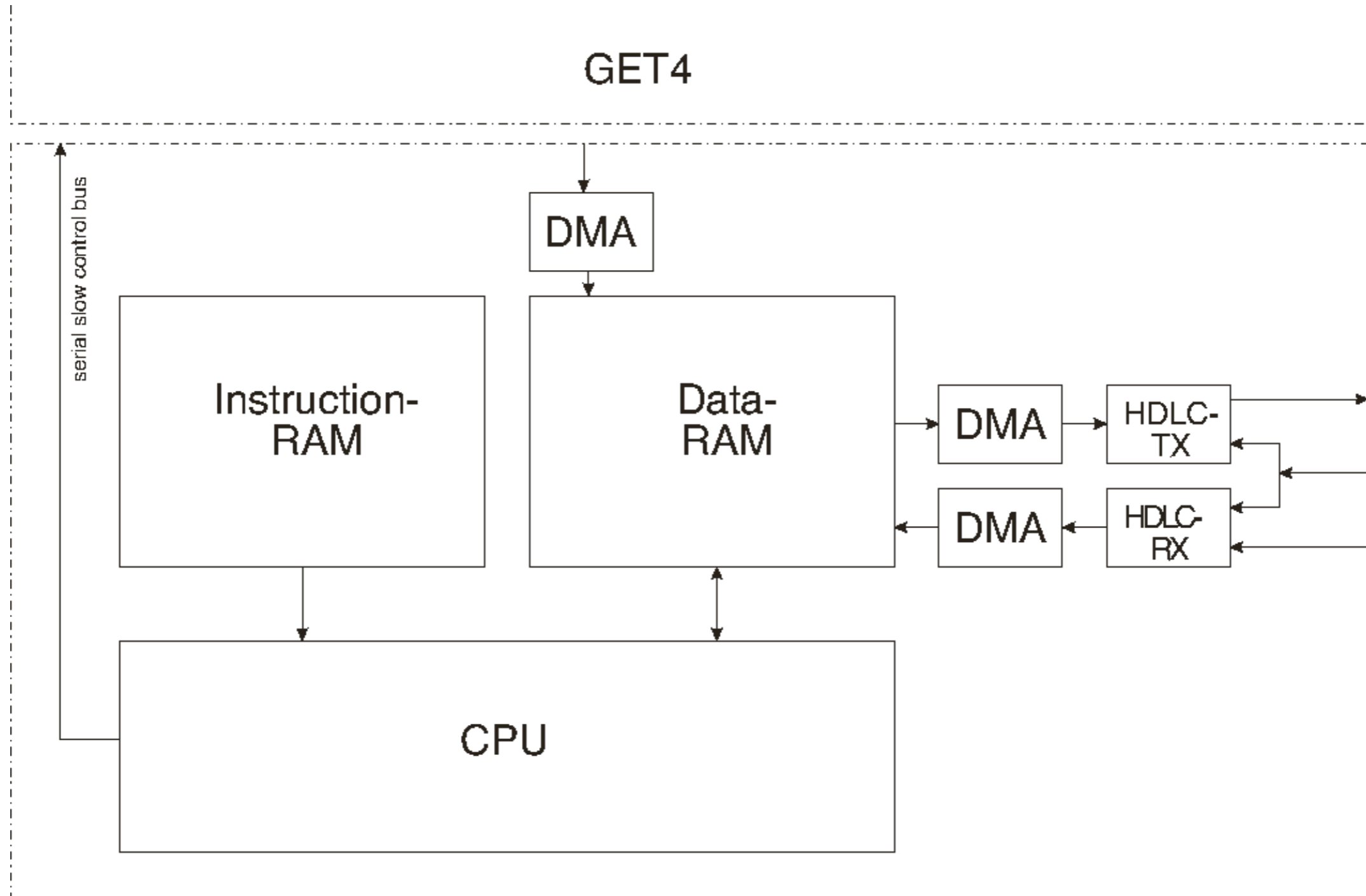
Chip-ID	Epoch-No
	Event 1
	Event 2
	Event 3
	Event 4
	Event 5

⋮

# Idee: Intelligenter Readout-ASIC

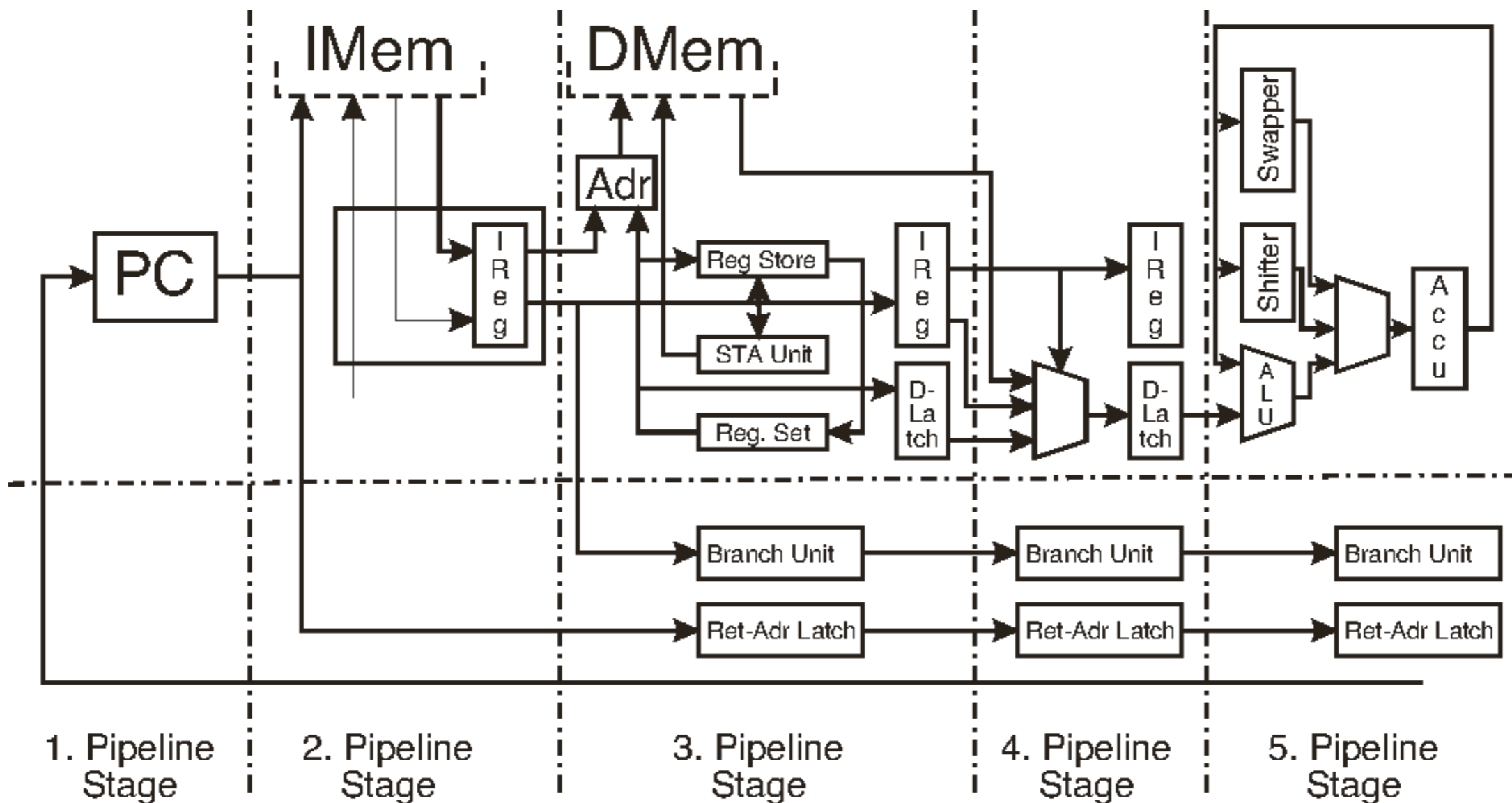
- Vorgesehene Funktionen
  - Pufferung der Event-Daten
  - Erstellen von Datenframes
  - Datentransport in einer Token-Ring Auslese
- Realisierung als separater Chip
  - Abhängig von den Erfahrungen später möglicherweise Implementierung auf dem GET4
- Testchip zusammen mit Slowcontrol Komponenten
  - DAC
  - Komparatoren für Regelspannungsüberwachung

# Idee: Intelligenter Readout-ASIC



# SEU harte CPU

- 32-Bit-CPU mit fünfstufiger Pipeline
- Alle Register, Speicher und Zustandsmaschinen sind mit Hamming-Codes gegen SEU gesichert



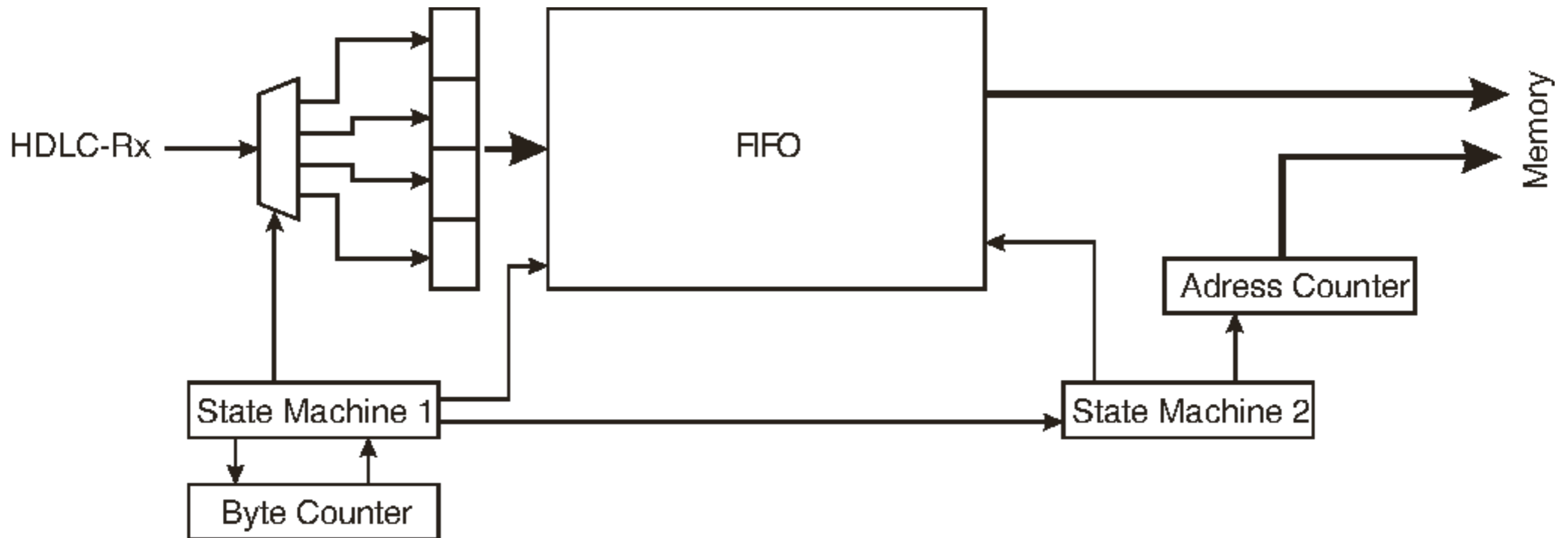
# SEE harte CPU

- 18 verschiedene Klassen von Maschinenbefehlen
- Mit jedem Taktzyklus wird ein Maschinenbefehl abgeschlossen
- 7 Interruptebenen

	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
ALU Op	0	0	Width		Type		2nd Operand														
Shift Op	0	1	Width		Type		X	X	X	X	X	X	X	X	X	X	X	X	X	X	
Swap	1	0	Width		0	Type		X	X	X	X	X	X	X	X	X	X	X	X	X	
Add to Register	1	0	X	X	1	1	0	0	0	Reg		Offset									
Load Register	1	0	X	X	1	1	0	0	1	Reg		Constant									
Store Akku	1	0	X	X	1	0	0	Address													
Load Akku	1	0	Width		1	0	1	Operand													
Status Bit Man.	1	0	X	X	1	1	0	1	0	0	X	X	X	X	Type		Bit				
Load Status Im.	1	0	X	X	1	1	0	1	0	1	X	X	X	X	X	Status					
Load Status	1	0	X	X	1	1	0	1	1	0	X	X	X	X	X	X	X	X	X	X	
Store Status	1	0	Width		1	1	0	1	1	1	X	X	X	X	X	X	X	X	X	X	
Load IO	1	0	Width		1	1	1	0	X	X	X	X	X	X	X	Address					
Store IO	1	0	Width		1	1	1	1	Byte Sel.		X	X	X	X	X	X	Address				
Jump	1	1	X	X	0	0	Condition				Address										
Jump Subroutine	1	1	X	X	0	1	Condition				Address										
Return	1	1	0	0	1	0	Condition				X	X	X	X	X	X	X	X	X	X	
Return from Int	1	1	0	1	1	0	Condition				X	X	X	X	X	X	X	X	X	X	
Sleep	1	1	X	X	1	1	Condition				X	X	X	X	X	X	X	X	X	X	

# DMA Kanäle

- Umwandlung von Bytes in 32-Bit-Worte
- FIFO zur Trennung von Clock-Domain
- Sobald FIFO gefüllt ist, werden Daten im Burst-Modus in Speicher geschrieben



# HDLC Übertragung

- Kein 8/10-Bit encoding: kein Overhead von 25 %
- Daten sind eingerahmt von Flags ("01111110")
- Zero-Insertion: Auf 5 aufeinanderfolgende Einsen wird eine Null eingefügt
- Cyclic Redundancy Check (CRC) zur Fehlererkennung

n octets





# Status

CPU	ok
Speicherarbitrierung	zu überarbeiten
DMA	ok
HDLC Transceiver	fehlt

- VHDL code für Teilmodule erstellt
- Simulation durchgeführt
- Synthetisierung und Place and Routing steht noch aus