Programmierung Paralleler Systeme

5. 1 Architektur der Fukitsu VPP300/700

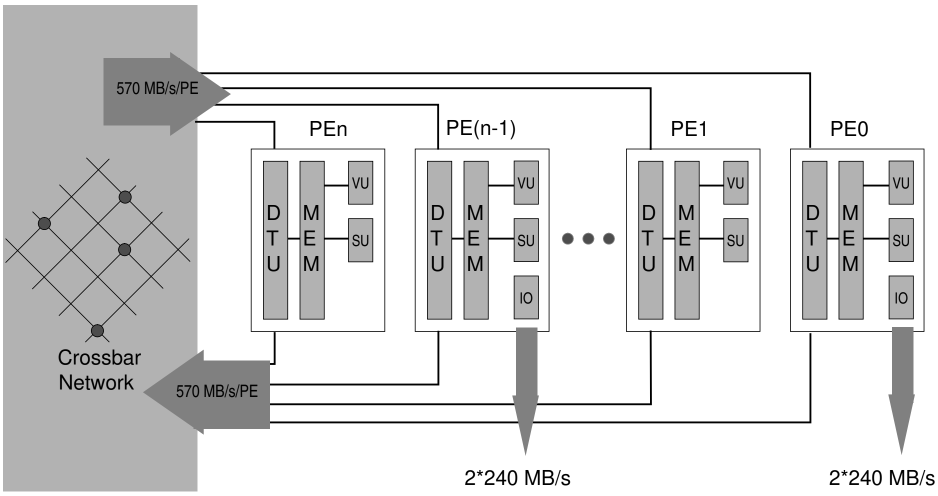
Credits: Fujitsu und SNI

Claus-Uwe Linster

IMMD IV, Lehrstuhl für Betriebssysteme
Universität Erlangen-Nürnberg

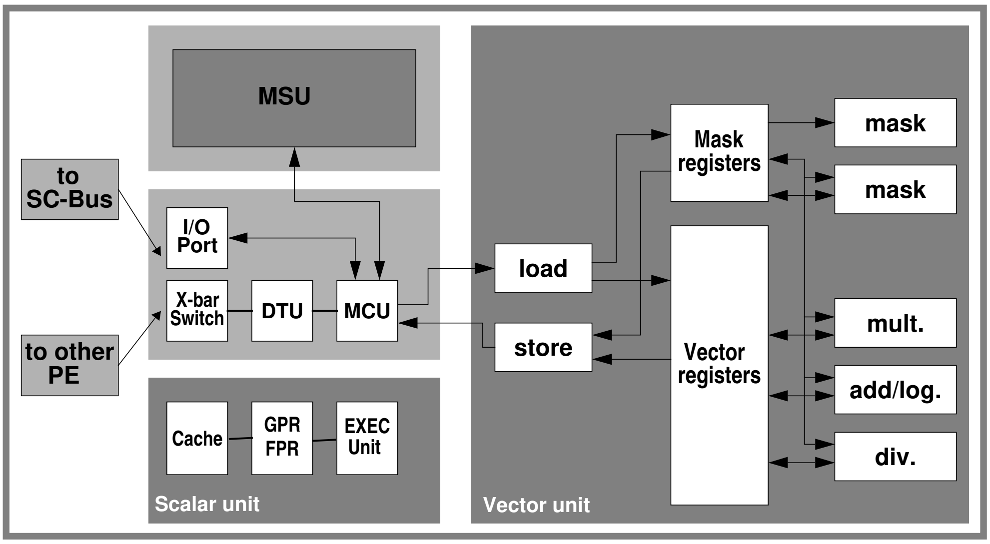
linster@informatik.uni-erlangen.de

5.1 Architektur

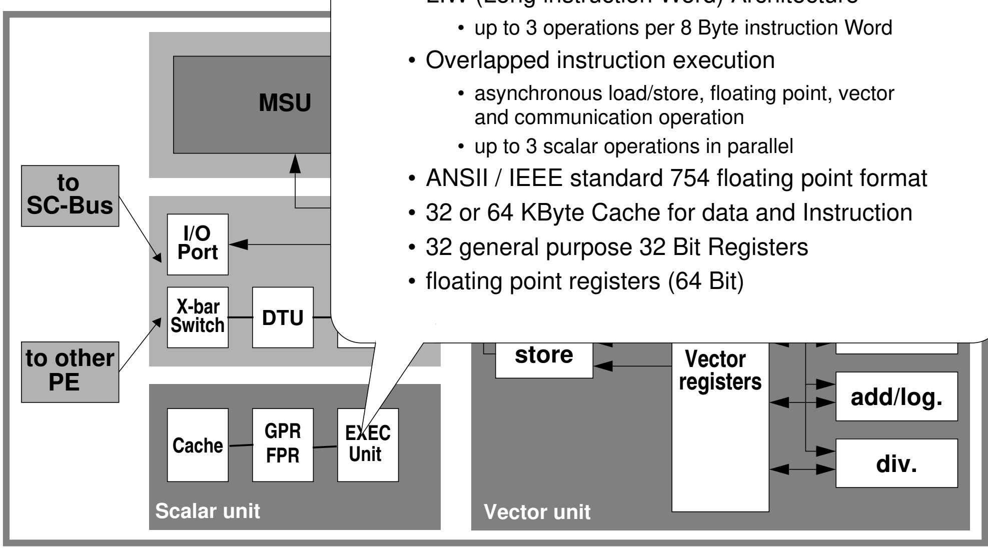


SU: Scalar Unit
VU: Vector Unit
IO: IO-Port
MEM:Memory
DTU: Data Transfer Unit

5.1.1 Prozessor



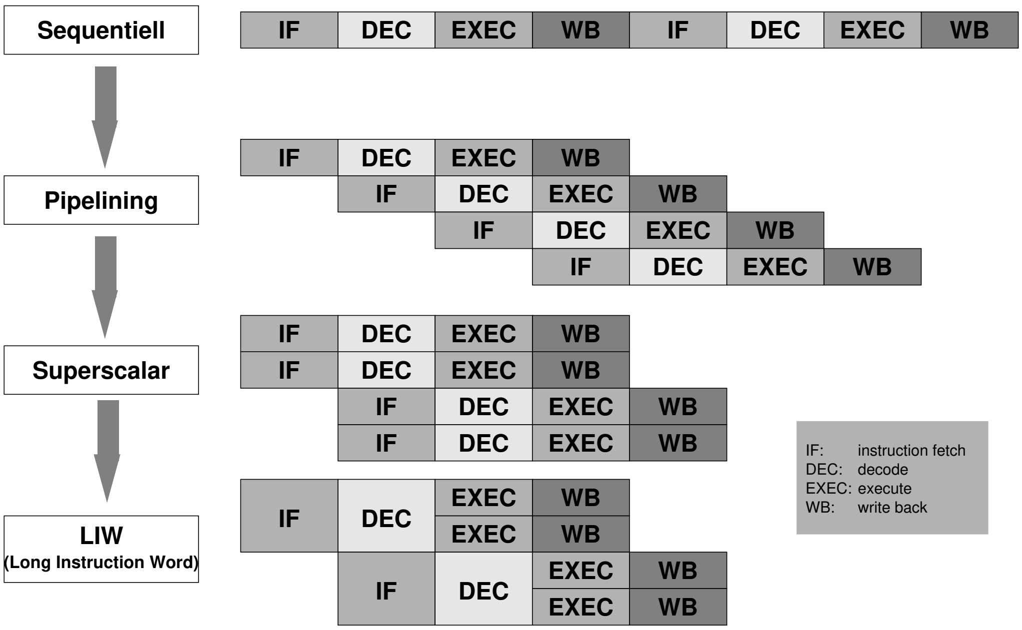
5.1.2 Scalar Unit Architecture



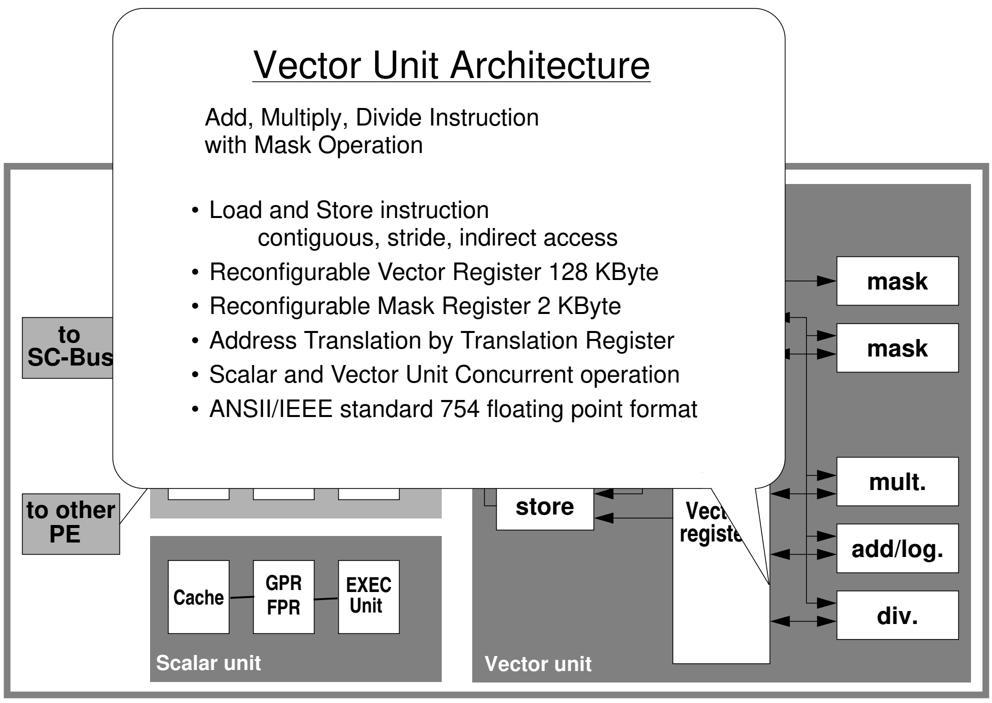
Scalar Unit Architecture

- RISC Scalar Unit
- LIW (Long instruction Word) Architecture
  - up to 3 operations per 8 Byte instruction Word
- Overlapped instruction execution
  - asynchronous load/store, floating point, vector and communication operation
  - up to 3 scalar operations in parallel
- ANSII / IEEE standard 754 floating point format
- 32 or 64 KByte Cache for data and Instruction
- 32 general purpose 32 Bit Registers
- floating point registers (64 Bit)

5.1.3 Long Instruction Word

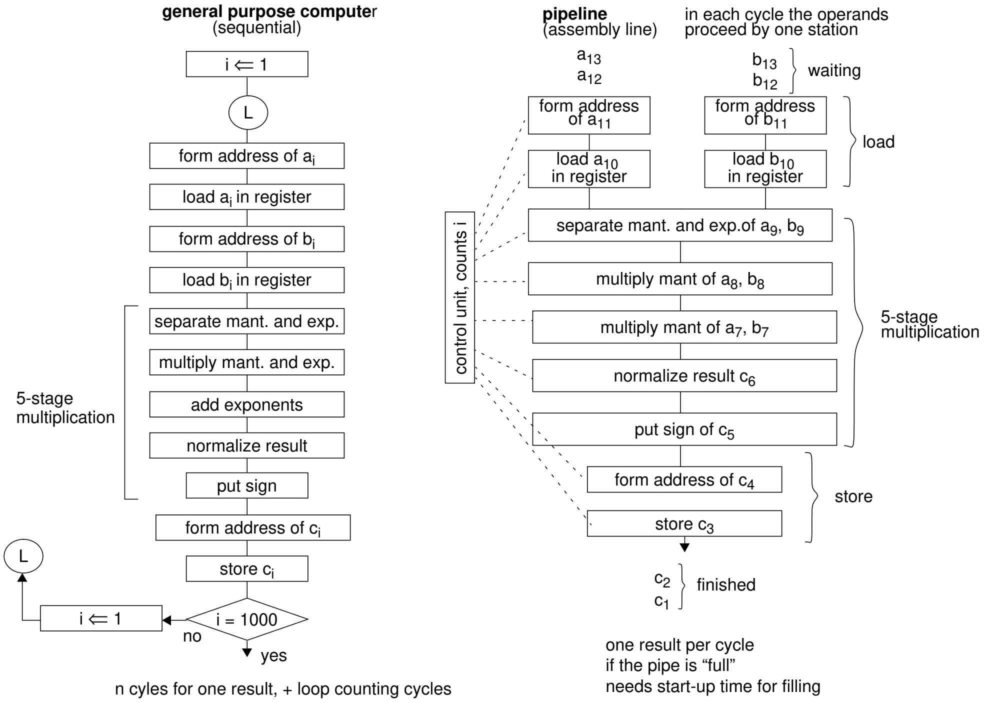


5.1.4 Vectoreinheit-Architektur



5.1.4.1 Instruction- vs. Vektorpipeline

▲ Problem: ci <== $a_i * b_i$



aus Willi Schönauer, 1998: “Scientific Supercomputing”: http: //www.uni-karlsruhe.de/~rz03/book/

5.1.4.2 VPP700 Vector Unit Peak Rates

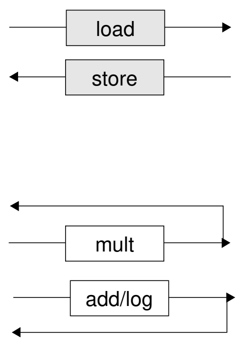
VPP700 Vector Unit Peak Rates

Load / Store Bandwidth

$$\frac{2x8x8Bytes}{7ns} = 18.2GBytes$$

Floating Point Peak Performance:

$$\frac{16FLOPS}{7ns} = 2.2GFLOPS$$



in general:

$$\frac{2N[Instr]}{N[cycles]} = Peak$$

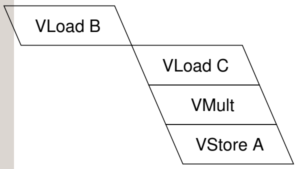
Taktzeit: 7 ns
Taktfrequenz 143MHz
Zugriffszeit Arbeitsspeicher: 60 ns; 512-fach interleaved
Max. Scalarleistung: 275 MFlops

5.1.4.3 Peak Rate Examples

Peak Rate Examples

Kernel:
A(I) = B(I)*C(I)

Instruction Diagram:

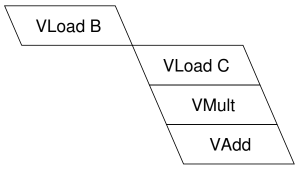


Performance:

$$\frac{N[FLOP]}{2N[cycles]} = \frac{1}{4}Peak$$

Kernel:
S = S + B(I)*C(I)

Instruction Diagram:

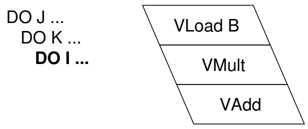


Performance:

$$\frac{2N[FLOP]}{2N[cycles]} = \frac{1}{2}Peak$$

Kernel:
A(I,J)=A(I,J)+B(I,K)*C(K,J)

Instruction Diagram:



Performance:

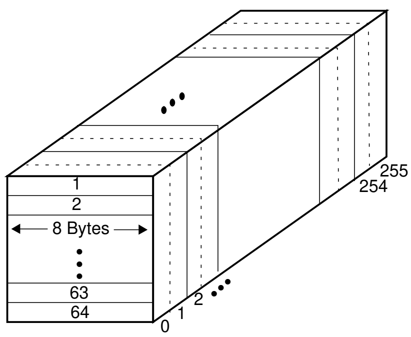
$$\frac{2N[FLOP]}{N[cycles]} = Peak$$

5.1.4.4 Vector Registers

Vector Registers

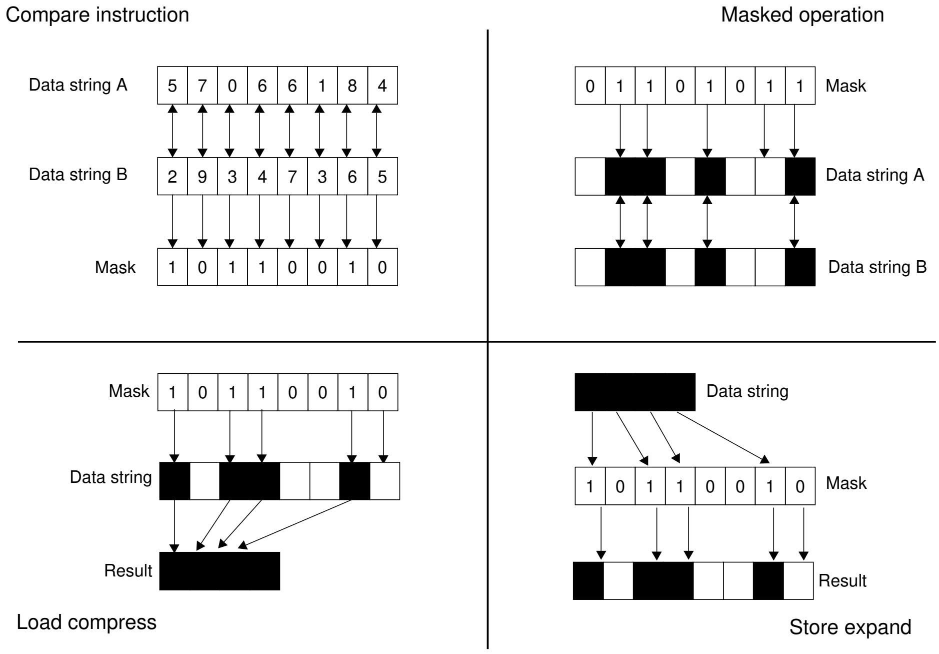
Total capacity: 128 KBytes

Number of registers	Size of registers
256	64
128	128
64	256
32	512
16	1024
8	2048

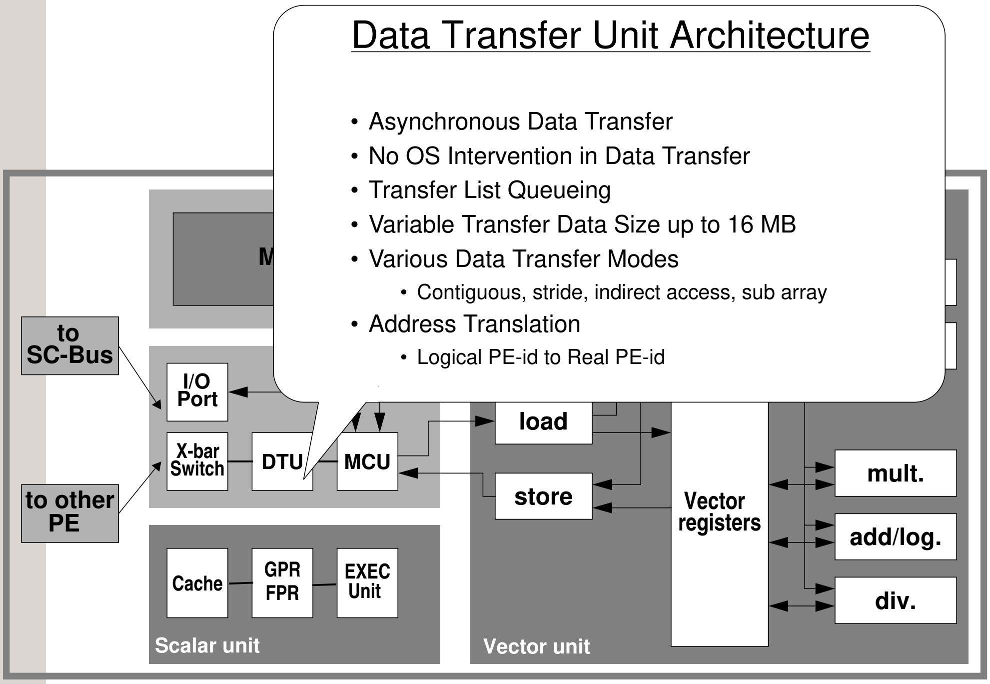


5.1.4.5 Mask Registers

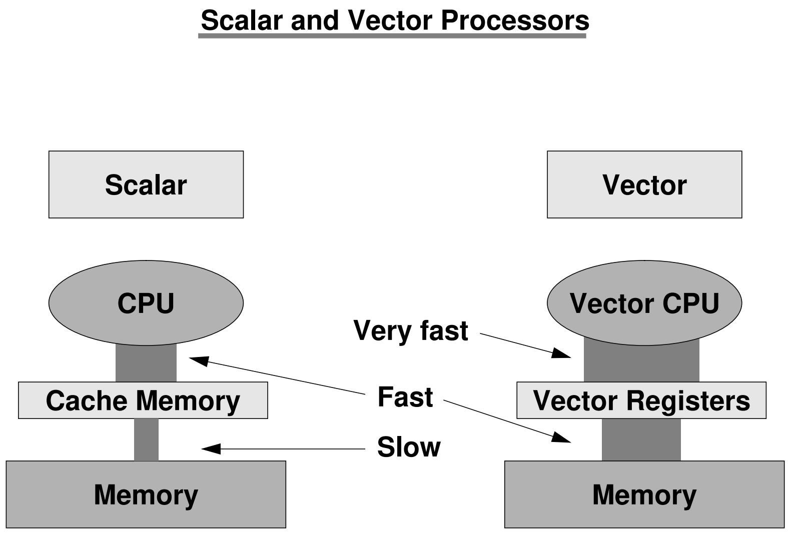
Mask Registers



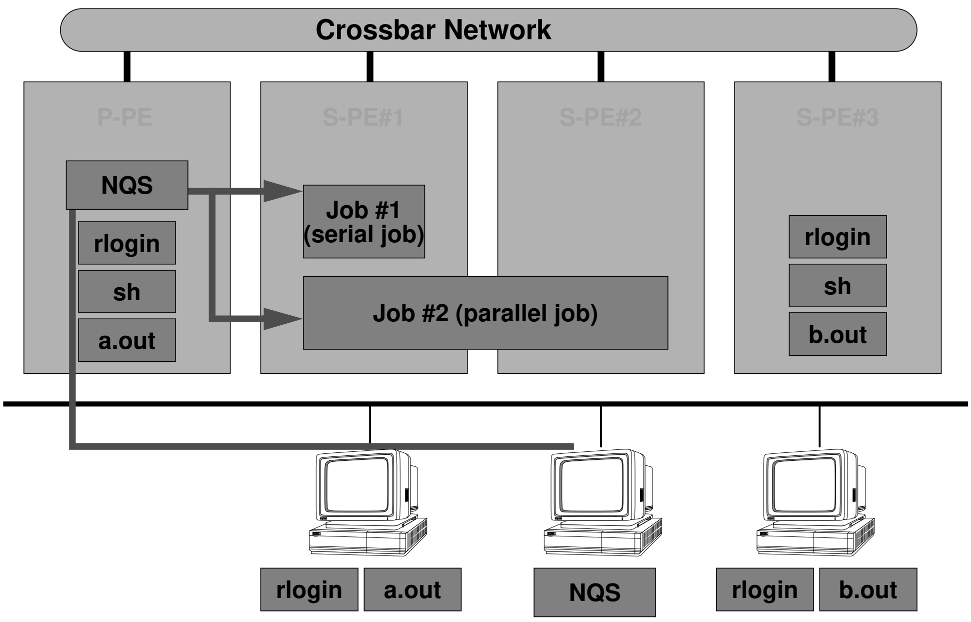
5.1.5 Data Transfer Unit



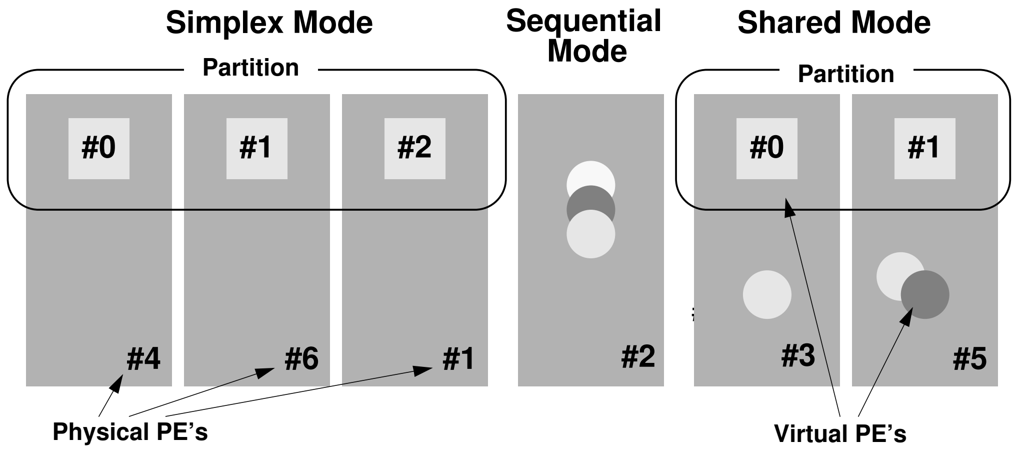
5.1.6 Speicher-Bandbreite



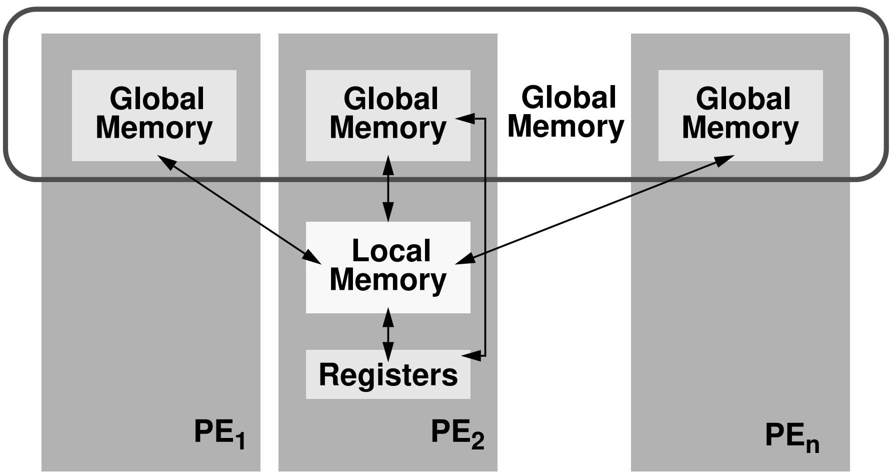
5.1.7 Auftragssteuerung



5.1.8 Betriebsmodi



5.1.9 Speicherzugriff



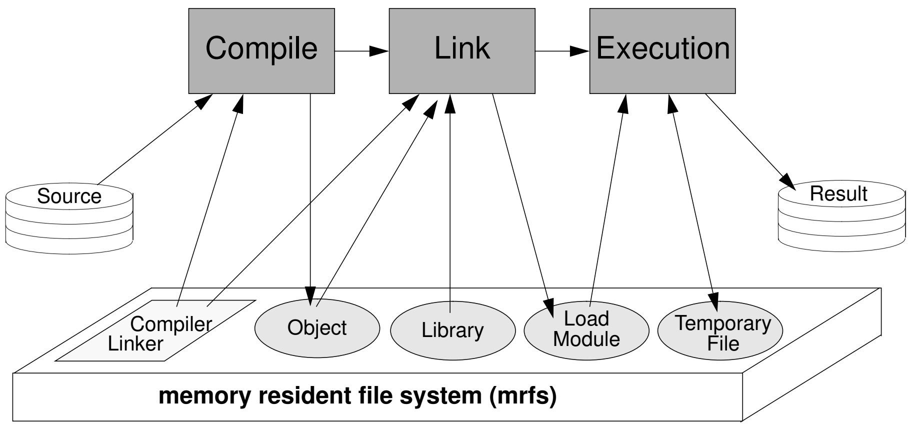
5.1.10 Software-Umgebung

Pre-installed

- **Programming**
  - **Compiler**
    - Fortran 90/VP
    - Fortran 90/VPP
    - C
    - C/VP
    - C++
  - **Message PassingLibraries**
    - Fortran 90/VP
    - Fortran 90/VPP
- **Programming Tools**
  - Analyzer
  - VPP Workbench
  - Engineering Development
- **Scientific Libraries**
  - SSLII/VP
  - SSLII/VPP

5.1.11 Memory resident File System

- Compile, link and execute programs without disk access
- Frequently used commands and libraries resident in memory



5.1.12 VPP-300/700 Konfigurationen

- Erlangen
  - 6 Prozessoren VPP300, 2 GByte
  - davon 1 Prozessor mit E/A; identisch mit Primary-PE
  - 65 GByte Plattenperipherie
  - Dialog: shared; pe0;
  - NQS-Queues
    - express/compile: 1h cpu; 512 MB; pe0
    - work: 30h cpu; 1800 MB; pe1
    - parallel: 30h cpu; 4 x 1544 MB; pe2 - pe5
- München (LRZ)
  - 52 Prozessoren VPP700, 2 GByte
  - davon 5 Prozessoren mit E/A + 1 Primary PE (Systemplatten; NQS)
  - 904 GByte Platenperipherie