Datein:

-TLE

-Package for Sizes and subtypes

-Package for mnemonics

-Package for Constants and symbolic register

-Package memory content

-Package auxillery functions

-Package for instructions encoding

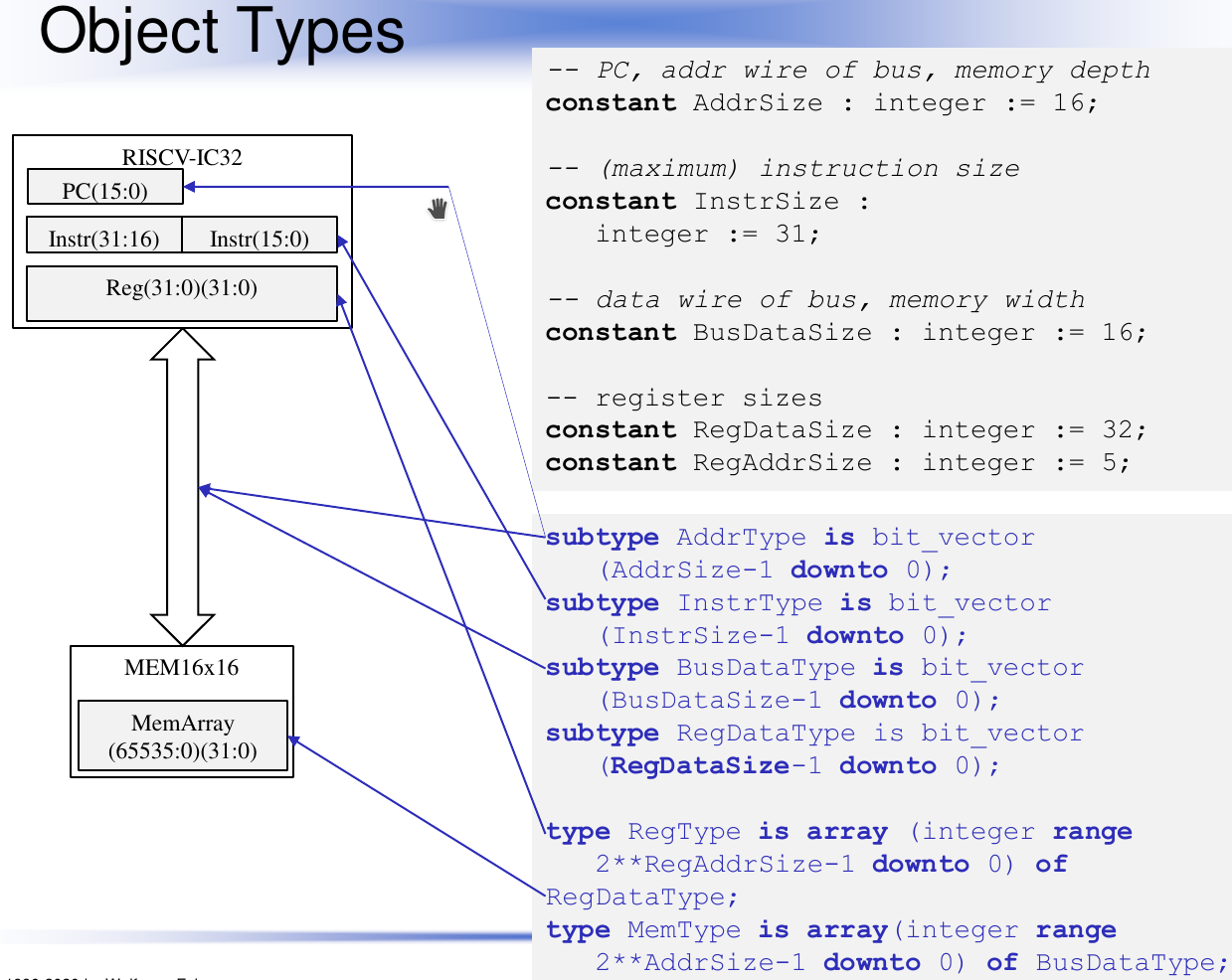
-Testbenche

# TLE

Simple TLE wie im erstens Projekt.

# Package for Sizes and Subtypes: Yu-Hung

Risc-V.PDF Seite.6



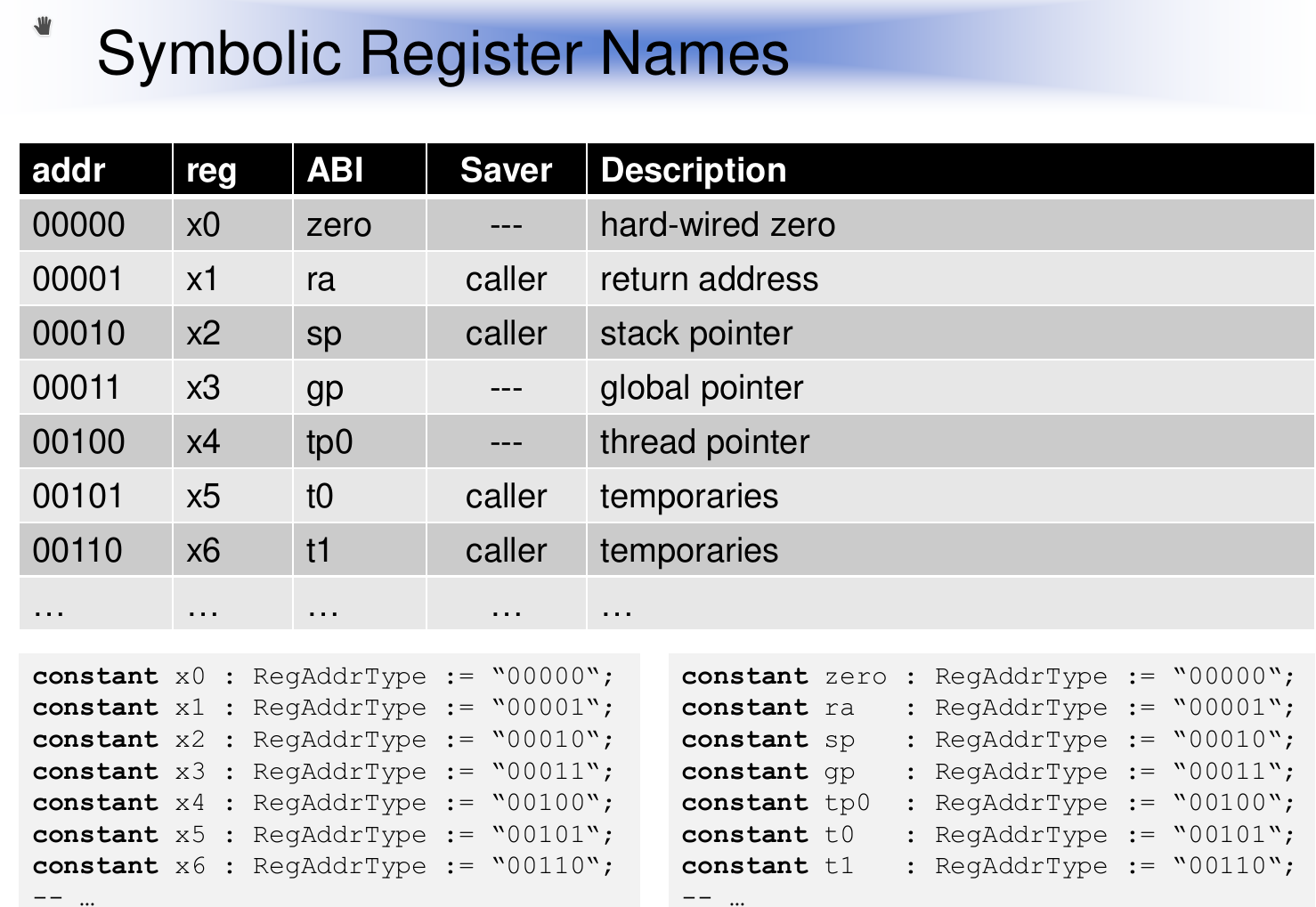
# Package for mnemonics

Risc-V.PDF Seite.8

# LB – 8 bit or ….? Depends on the lecture

# Package for Constants and symbolic register: Arun

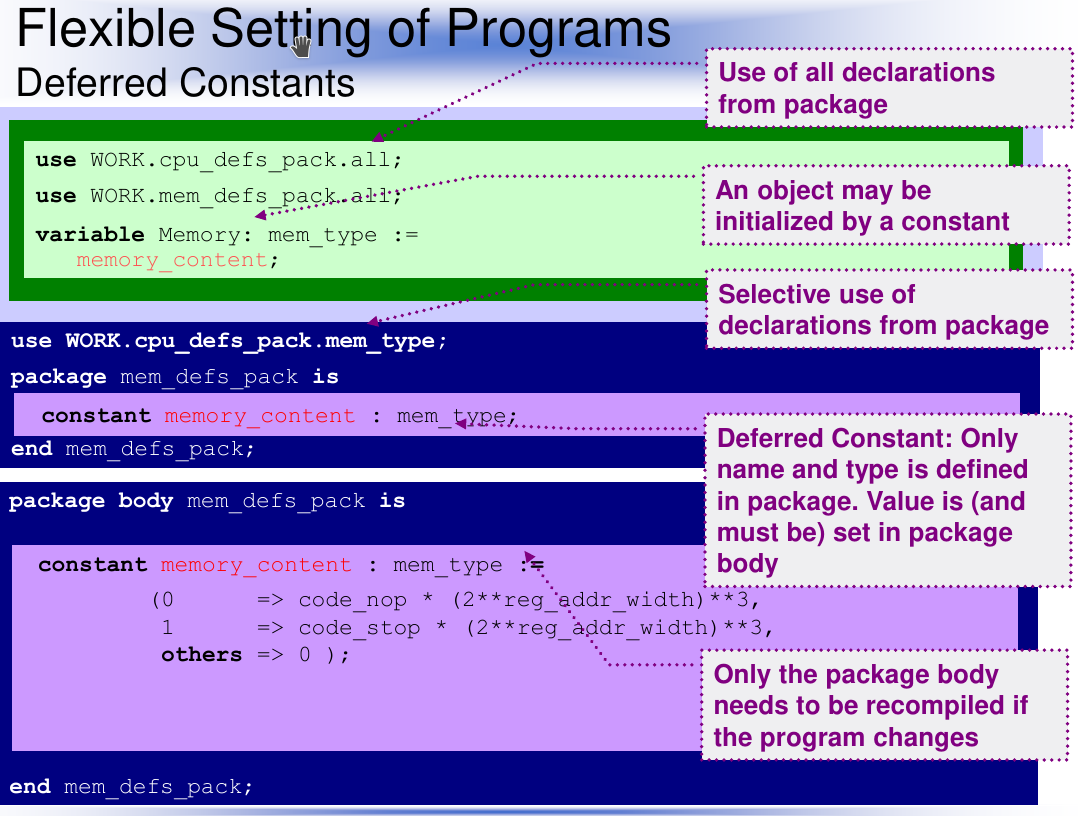
Risc-V.PDF Seite.35



Unsere Symbolic Register müssen als Integer angegeben warden, weil Reg Addressen als Integer definiert sind.

# -Package memory content: Voon

P03-1 BasicRegisterOperation.PDF Seite.25

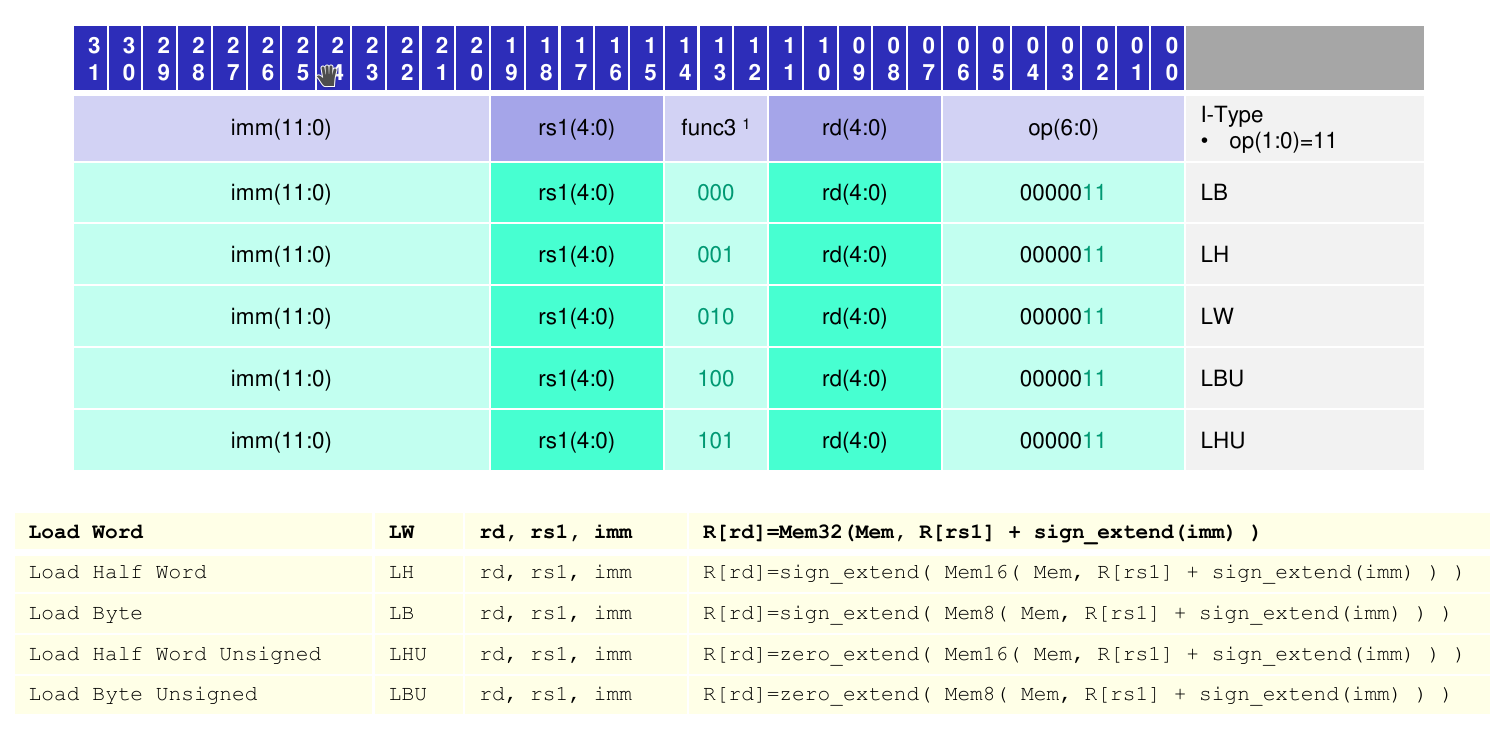
Hardcoded Inhalt von Mem.

# -Package auxillery functions

Conversions to string or formating string (auch bit-String gemeint)

# Package for instruction decoding: Tiemo

Risc-V.pdf Seite.10

Jump Instruction:

-Risc v spec S.38

-JAL: (unconditional jump)

1. Just jump to destination register rd: PC := Mem(rd)
2. No return possible with this Instruction

-JALR (Jump with return address)

1. Save address after jump Instruction to rd(often Register x1)
2. Add imm to rs1
3. Jump to rs1+imm (something about LSB to 0, don’t understand): PC := Mem(rs1+imm)

-Arithmetic Operations

-Risc v spec S.37

-ADD

-ADDI

-SUB

-Logical Instructions

-Risc v spec S.37

-AND

-OR

-XOR

-XORI

-ORI

-ANDI

# Testbench: Voon

Testbench to test Cornercases.

