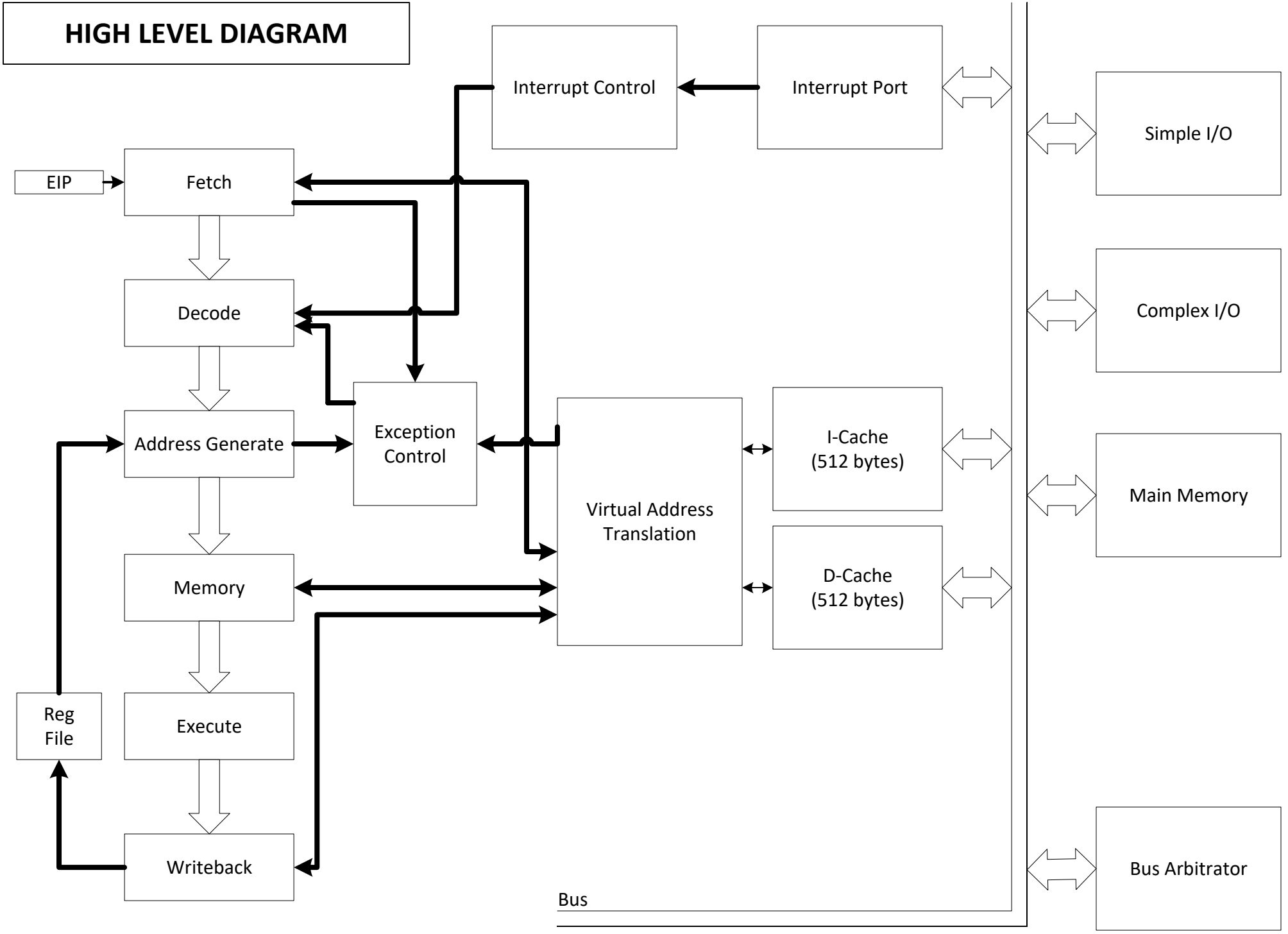
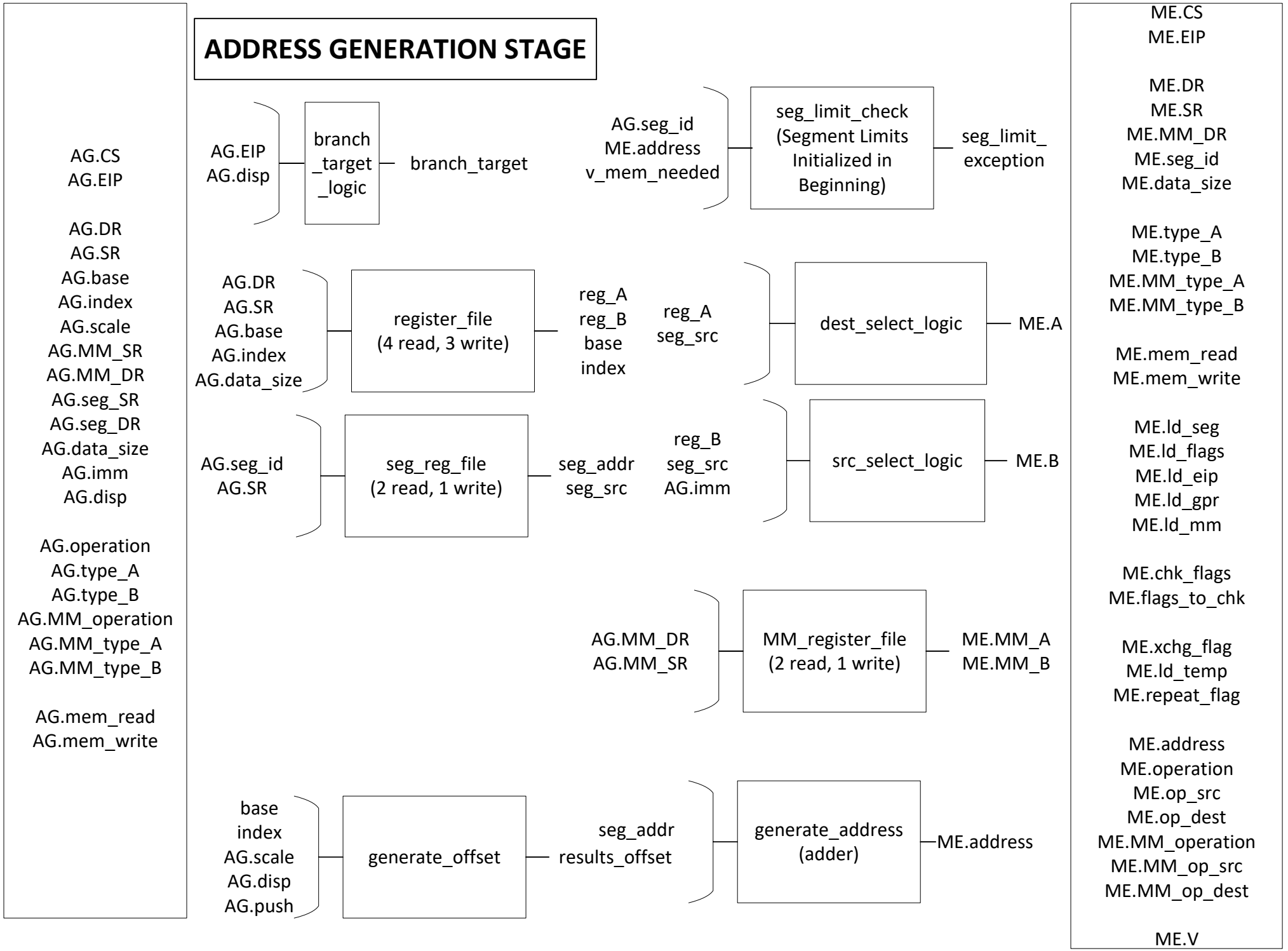
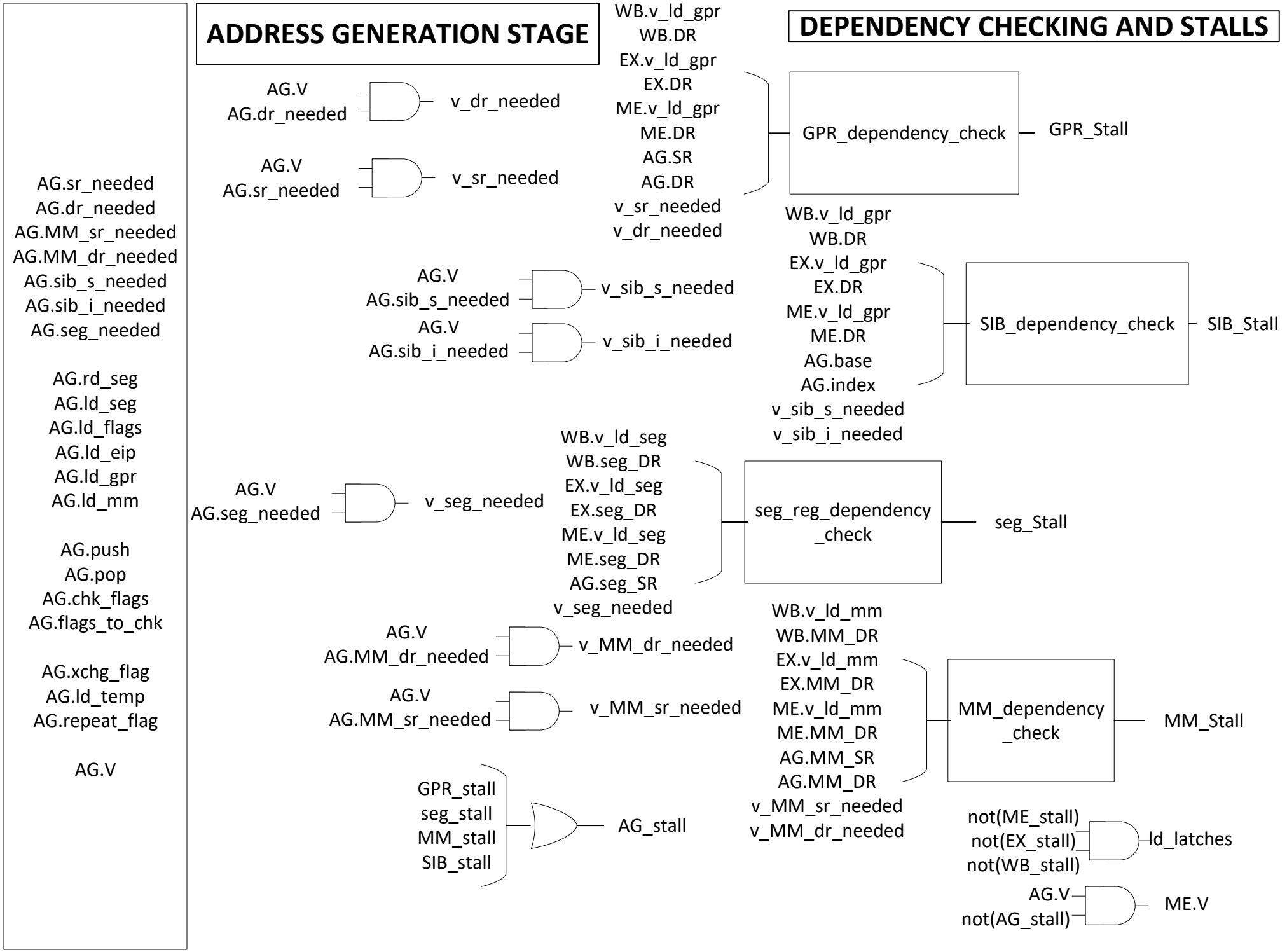
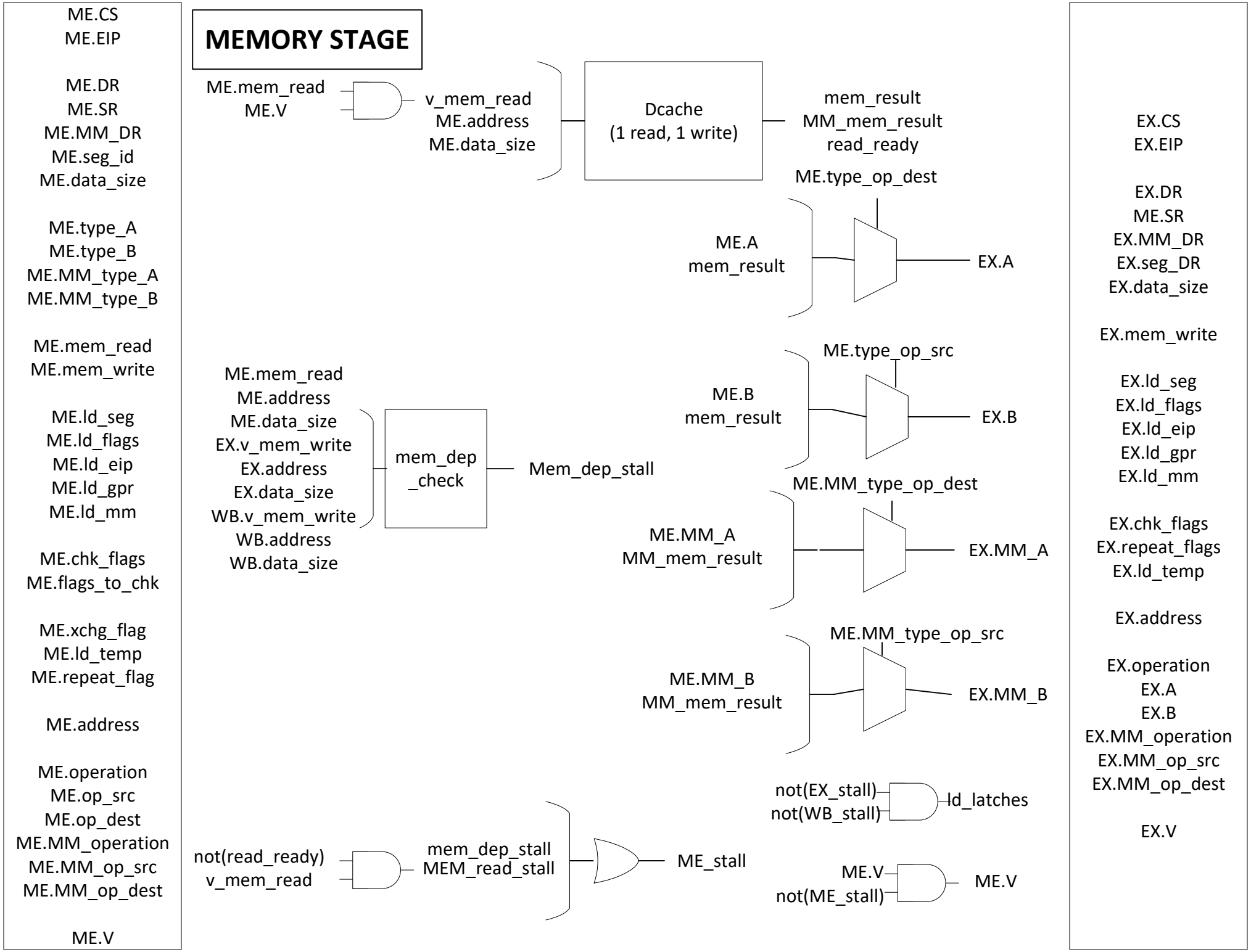
HIGH LEVEL DIAGRAM




ADDRESS GENERATION STAGE
AG.CS
AG.EIP
AG.DR
AG.SR
AG.base
AG.index
AG.scale
AG.MM_SR
AG.MM_DR
AG.seg_SR
AG.seg_DR
AG.data_size
AG.imm
AG.disp
AG.operation
AG.type_A
AG.type_B
AG.MM_operation
AG.MM_type_A
AG.MM_type_B
AG.mem_read
AG.mem_write
AG.EIP
AG.disp
branch _target _logic
branch_target
AG.seg_id
ME.address
v_mem_needed
seg_limit_check (Segment Limits Initialized in Beginning)
seg_limit_ exception
AG.DR
AG.SR
AG.base
AG.index
AG.data_size
register_file (4 read, 3 write)
reg_A
reg_B
base
index
reg_A
seg_src
dest_select_logic
ME.A
AG.seg_id
AG.SR
seg_reg_file (2 read, 1 write)
seg_addr
seg_src
reg_B
seg_src
AG.imm
src_select_logic
ME.B
AG.MM_DR
AG.MM_SR
MM_register_file (2 read, 1 write)
ME.MM_A
ME.MM_B
base
index
AG.scale
AG.disp
AG.push
generate_offset
seg_addr
results_offset
generate_address (adder)
ME.address
ME.CS
ME.EIP
ME.DR
ME.SR
ME.MM_DR
ME.seg_id
ME.data_size
ME.type_A
ME.type_B
ME.MM_type_A
ME.MM_type_B
ME.mem_read
ME.mem_write
ME.ld_seg
ME.ld_flags
ME.ld_eip
ME.ld_gpr
ME.ld_mm
ME.chk_flags
ME.flags_to_chk
ME.xchg_flag
ME.ld_temp
ME.repeat_flag
ME.address
ME.operation
ME.op_src
ME.op_dest
ME.MM_operation
ME.MM_op_src
ME.MM_op_dest
ME.V


ADDRESS GENERATION STAGE
DEPENDENCY CHECKING AND STALLS
AG.sr_needed
AG.dr_needed
AG.MM_sr_needed
AG.MM_dr_needed
AG.sib_s_needed
AG.sib_i_needed
AG.seg_needed
AG.rd_seg
AG.ld_seg
AG.ld_flags
AG.ld_eip
AG.ld_gpr
AG.ld_mm
AG.push
AG.pop
AG.chk_flags
AG.flags_to_chk
AG.xchg_flag
AG.ld_temp
AG.repeat_flag
AG.V
AG.V
AG.dr_needed
v_dr_needed
AG.V
AG.sr_needed
v_sr_needed
WB.v_ld_gpr
WB.DR
EX.v_ld_gpr
EX.DR
ME.v_ld_gpr
ME.DR
AG.SR
AG.DR
v_sr_needed
v_dr_needed
GPR_dependency_check
GPR_Stall
AG.V
AG.sib_s_needed
v_sib_s_needed
AG.V
AG.sib_i_needed
v_sib_i_needed
WB.v_ld_gpr
WB.DR
EX.v_ld_gpr
EX.DR
ME.v_ld_gpr
ME.DR
AG.base
AG.index
v_sib_s_needed
v_sib_i_needed
SIB_dependency_check
SIB_Stall
AG.V
AG.seg_needed
v_seg_needed
WB.v_ld_seg
WB.seg_DR
EX.v_ld_seg
EX.seg_DR
ME.v_ld_seg
ME.seg_DR
AG.seg_SR
v_seg_needed
seg_reg_dependency
_check
seg_Stall
AG.V
AG.MM_dr_needed
v_MM_dr_needed
AG.V
AG.MM_sr_needed
v_MM_sr_needed
WB.v_ld_mm
WB.MM_DR
EX.v_ld_mm
EX.MM_DR
ME.v_ld_mm
ME.MM_DR
AG.MM_SR
AG.MM_DR
v_MM_sr_needed
v_MM_dr_needed
MM_dependency
_check
MM_Stall
GPR_stall
seg_stall
MM_stall
SIB_stall
AG_stall
not(ME_stall)
not(EX_stall)
not(WB_stall)
ld_latches
AG.V
not(AG_stall)
ME.V


ME.CS
ME.EIP
ME.DR
ME.SR
ME.MM_DR
ME.seg_id
ME.data_size
ME.type_A
ME.type_B
ME.MM_type_A
ME.MM_type_B
ME.mem_read
ME.mem_write
ME.ld_seg
ME.ld_flags
ME.ld_eip
ME.ld_gpr
ME.ld_mm
ME.chk_flags
ME.flags_to_chk
ME.xchg_flag
ME.ld_temp
ME.repeat_flag
ME.address
ME.operation
ME.op_src
ME.op_dest
ME.MM_operation
ME.MM_op_src
ME.MM_op_dest
ME.V
MEMORY STAGE
ME.mem_read
ME.V
v_mem_read
ME.address
ME.data_size
Dcache
(1 read, 1 write)
mem_result
MM_mem_result
read_ready
ME.type_op_dest
ME.A
mem_result
EX.A
ME.type_op_src
ME.B
mem_result
EX.B
ME.mem_read
ME.address
ME.data_size
EX.v_mem_write
EX.address
EX.data_size
WB.v_mem_write
WB.address
WB.data_size
mem_dep
_check
Mem_dep_stall
ME.MM_type_op_dest
ME.MM_A
MM_mem_result
EX.MM_A
ME.MM_type_op_src
ME.MM_B
MM_mem_result
EX.MM_B
not(EX_stall)
not(WB_stall)
ld_latches
not(read_ready)
v_mem_read
mem_dep_stall
MEM_read_stall
ME_stall
ME.V
not(ME_stall)
ME.V
EX.CS
EX.EIP
EX.DR
ME.SR
EX.MM_DR
EX.seg_DR
EX.data_size
EX.mem_write
EX.ld_seg
EX.ld_flags
EX.ld_eip
EX.ld_gpr
EX.ld_mm
EX.chk_flags
EX.repeat_flags
EX.ld_temp
EX.address
EX.operation
EX.A
EX.B
EX.MM_operation
EX.MM_op_src
EX.MM_op_dest
EX.V

EXECUTE STAGE

EX.EIP

EX.DR
EX.SR
EX.MM_DR
EX.seg_DR
EX.data_size

EX.mem_write

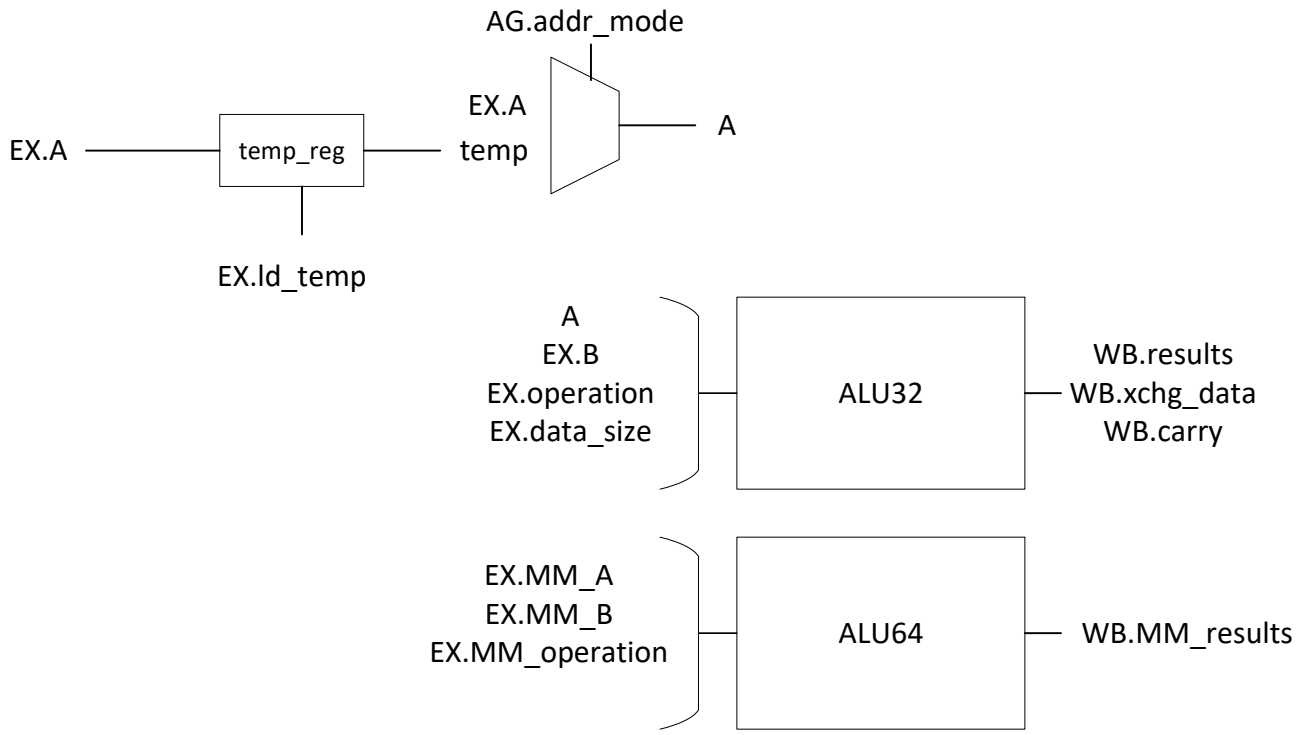
EX.ld_seg
EX.ld_flags
EX.ld_eip
EX.ld_gpr
EX.ld_mm

EX.chk_flags
EX.ld_temp
EX.repeat_flags
EX.operation
EX.MM_operation

EX.address

EX.A
EX.B
EX.MM_operation
EX.MM_A
EX.MM_B

EX.V



WB.EIP

WB.DR
WB.SR
WB.MM_DR
WB.seg_DR
WB.data_size

WB.mem_write

WB.ld_seg
WB.ld_flags
WB.ld_eip
WB.ld_gpr
WB.ld_mm

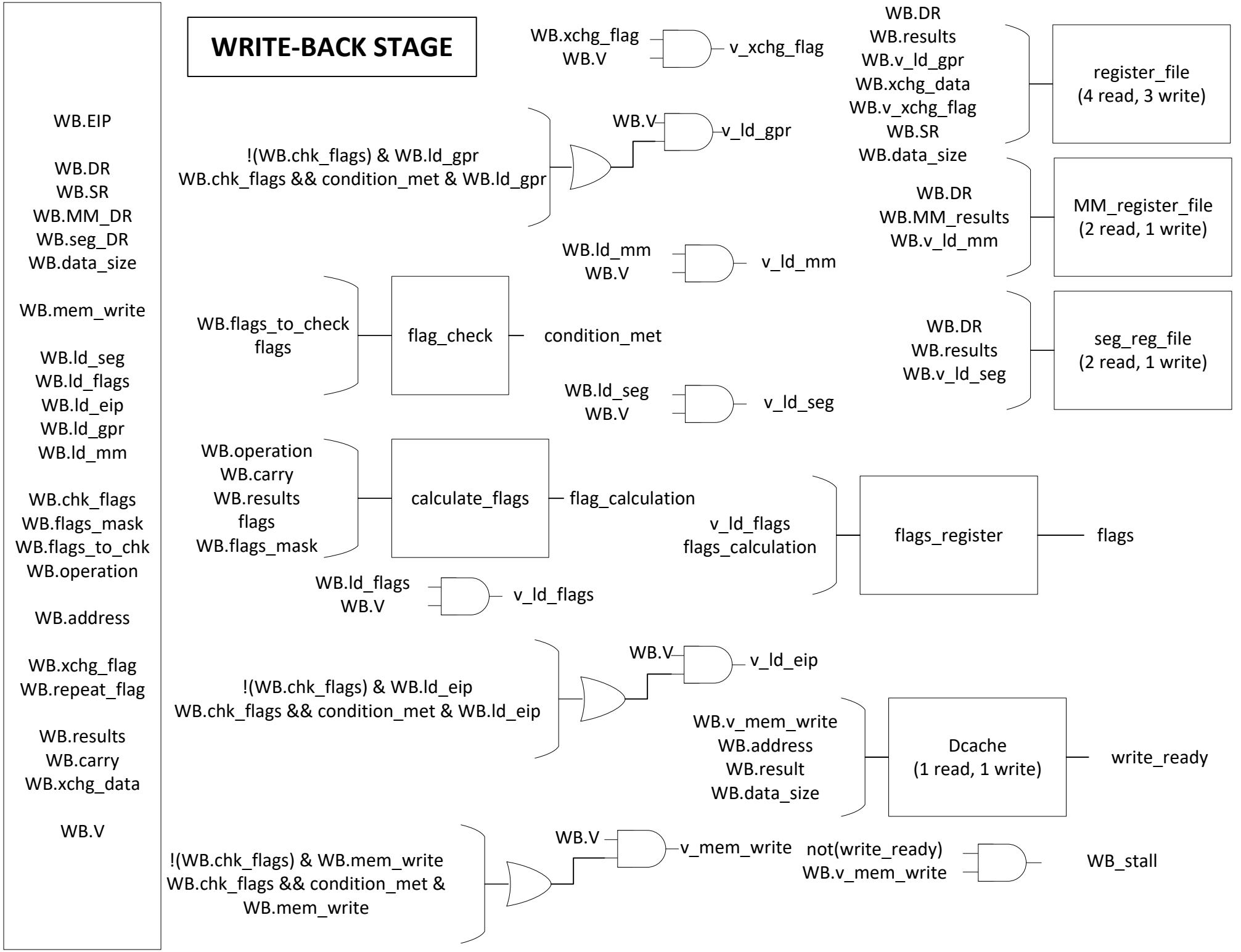
WB.chk_flags
WB.flags_to_chk

WB.address

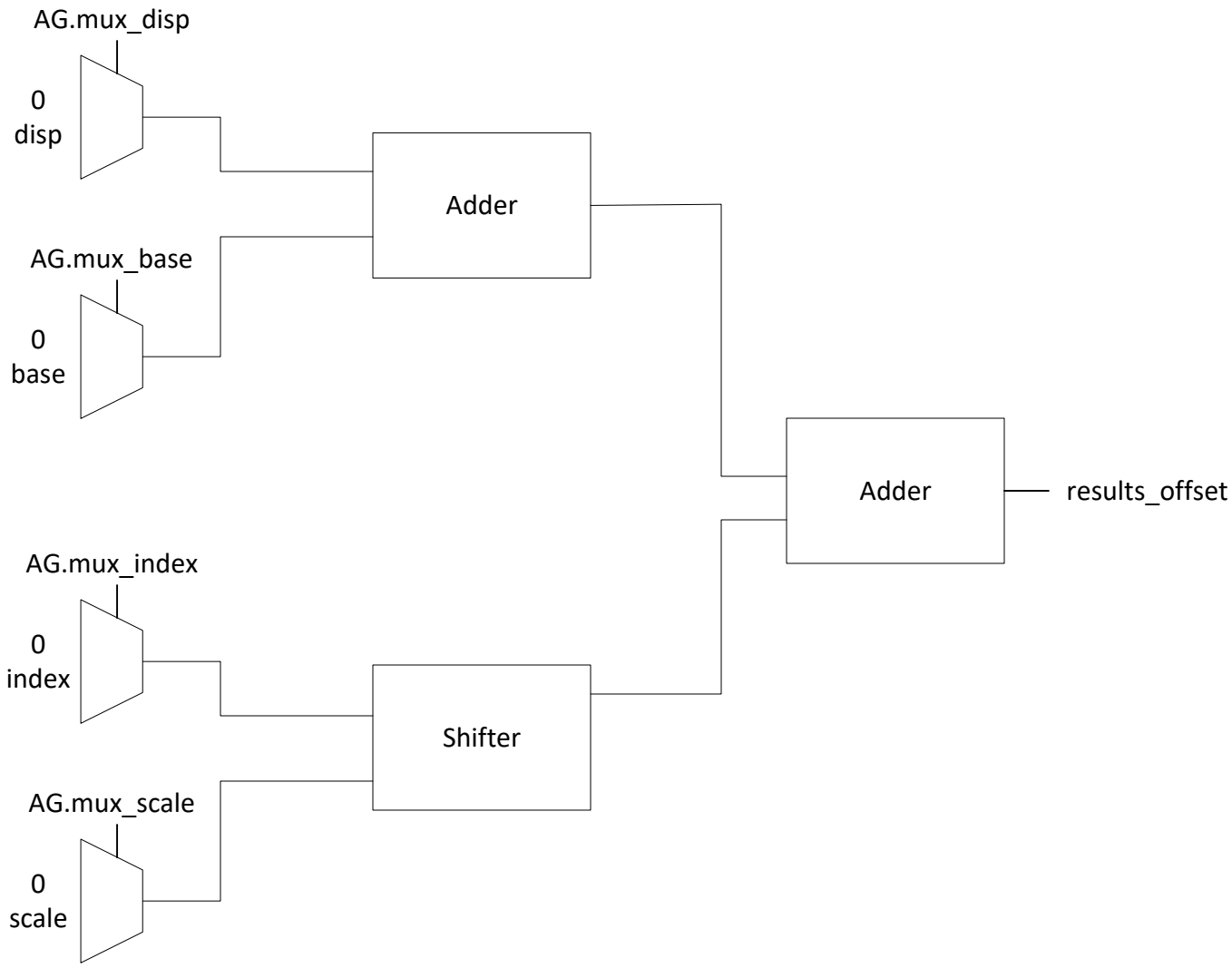
WB.xchg_flag
WB.repeat_flag

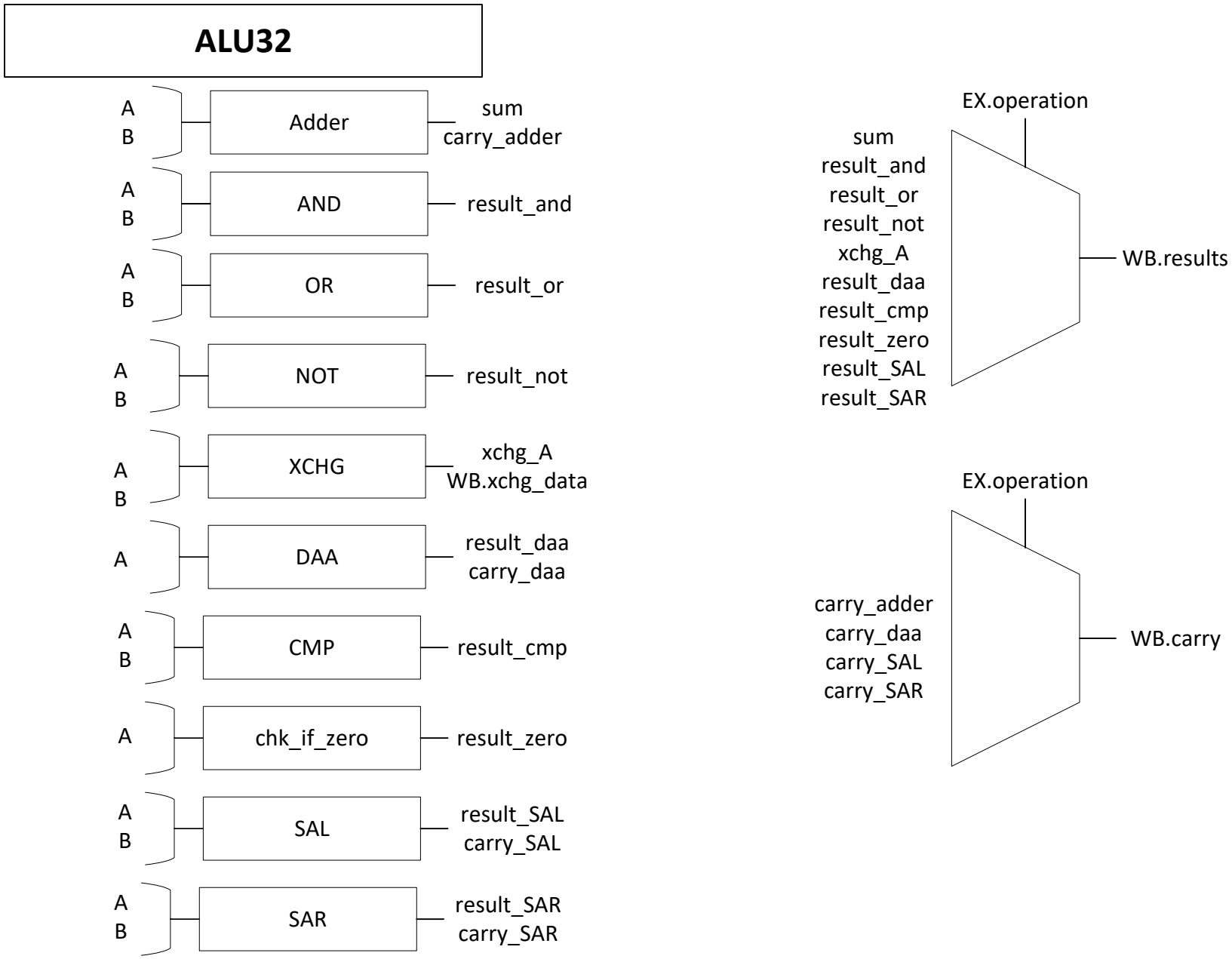
WB.results
WB.carry
WB.xchg_data

WB.V

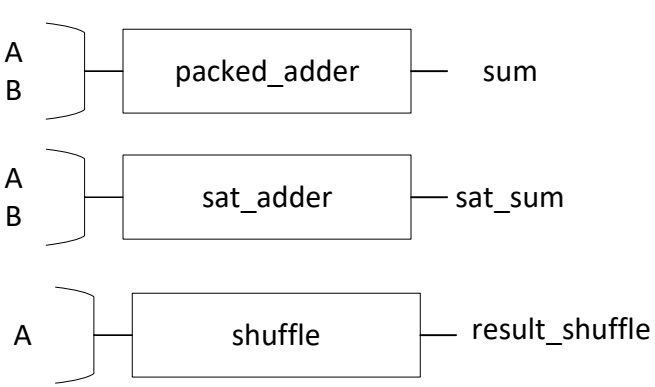

WB.EIP
WB.DR
WB.SR
WB.MM_DR
WB.seg_DR
WB.data_size
WB.mem_write
WB.ld_seg
WB.ld_flags
WB.ld_eip
WB.ld_gpr
WB.ld_mm
WB.chk_flags
WB.flags_mask
WB.flags_to_chk
WB.operation
WB.address
WB.xchg_flag
WB.repeat_flag
WB.results
WB.carry
WB.xchg_data
WB.V
WRITE-BACK STAGE
WB.xchg_flag
WB.V
v_xchg_flag
WB.DR
WB.results
WB.v_ld_gpr
WB.xchg_data
WB.v_xchg_flag
WB.SR
WB.data_size
register_file
(4 read, 3 write)
!(WB.chk_flags) & WB.ld_gpr
WB.chk_flags && condition_met & WB.ld_gpr
WB.V
v_ld_gpr
WB.DR
WB.MM_results
WB.v_ld_mm
MM_register_file
(2 read, 1 write)
WB.ld_mm
WB.V
v_ld_mm
WB.flags_to_check
flags
flag_check
condition_met
WB.DR
WB.results
WB.v_ld_seg
seg_reg_file
(2 read, 1 write)
WB.ld_seg
WB.V
v_ld_seg
WB.operation
WB.carry
WB.results
flags
WB.flags_mask
calculate_flags
flag_calculation
v_ld_flags
flags_calculation
flags_register
flags
WB.ld_flags
WB.V
v_ld_flags
!(WB.chk_flags) & WB.ld_eip
WB.chk_flags && condition_met & WB.ld_eip
WB.V
v_ld_eip
WB.v_mem_write
WB.address
WB.result
WB.data_size
Dcache
(1 read, 1 write)
write_ready
!(WB.chk_flags) & WB.mem_write
WB.chk_flags && condition_met &
WB.mem_write
WB.V
v_mem_write
not(write_ready)
WB.v_mem_write
WB_stall

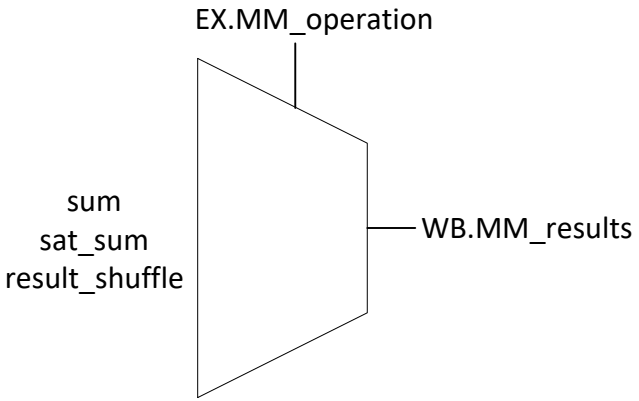
generate_offset

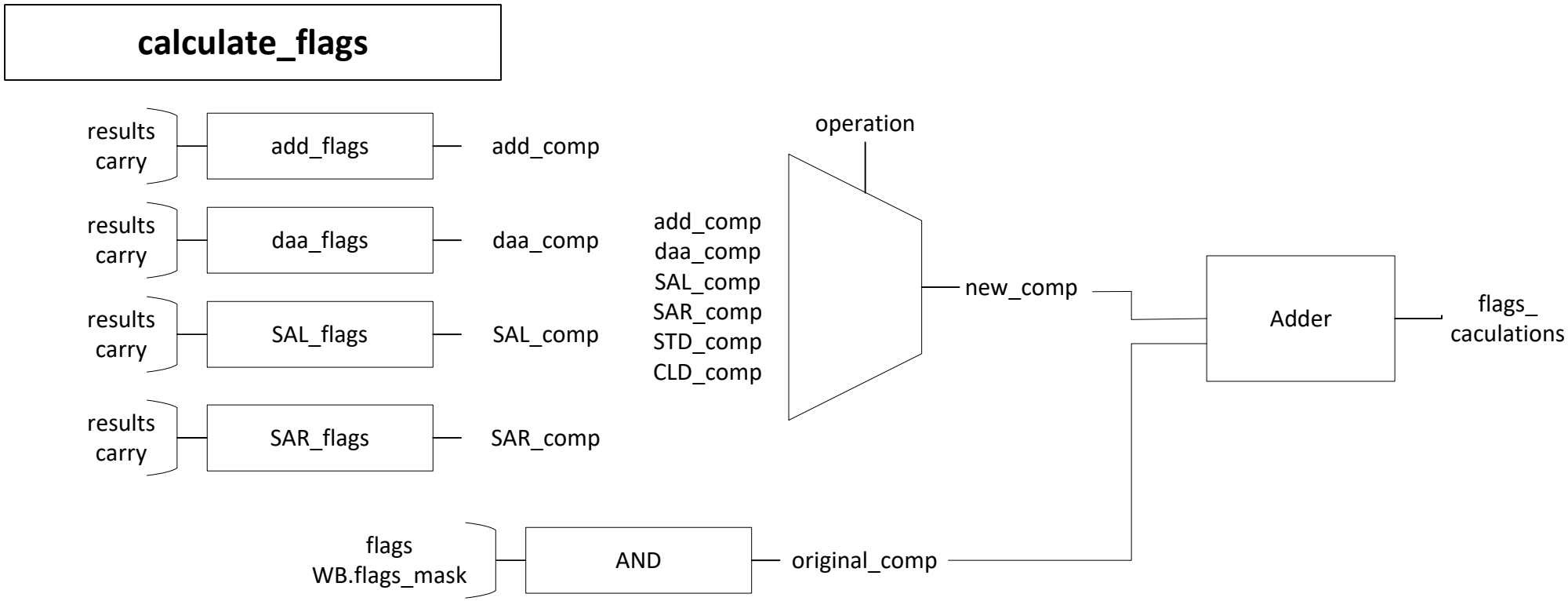



ALU32
A
B
Adder
sum
carry_adder
A
B
AND
result_and
A
B
OR
result_or
A
B
NOT
result_not
A
B
XCHG
xchg_A
WB.xchg_data
A
DAA
result_daa
carry_daa
A
B
CMP
result_cmp
A
chk_if_zero
result_zero
A
B
SAL
result_SAL
carry_SAL
A
B
SAR
result_SAR
carry_SAR
EX.operation
sum
result_and
result_or
result_not
xchg_A
result_daa
result_cmp
result_zero
result_SAL
result_SAR
WB.results
EX.operation
carry_adder
carry_daa
carry_SAL
carry_SAR
WB.carry

ALU64


A
B
packed_adder
sum
A
B
sat_adder
sat_sum
A
shuffle
result_shuffle


EX.MM_operation
sum
sat_sum
result_shuffle
WB.MM_results


calculate_flags
results
carry
add_flags
add_comp
results
carry
daa_flags
daa_comp
results
carry
SAL_flags
SAL_comp
results
carry
SAR_flags
SAR_comp
operation
add_comp
daa_comp
SAL_comp
SAR_comp
STD_comp
CLD_comp
new_comp
Adder
flags_
caculations
flags
WB.flags_mask
AND
original_comp