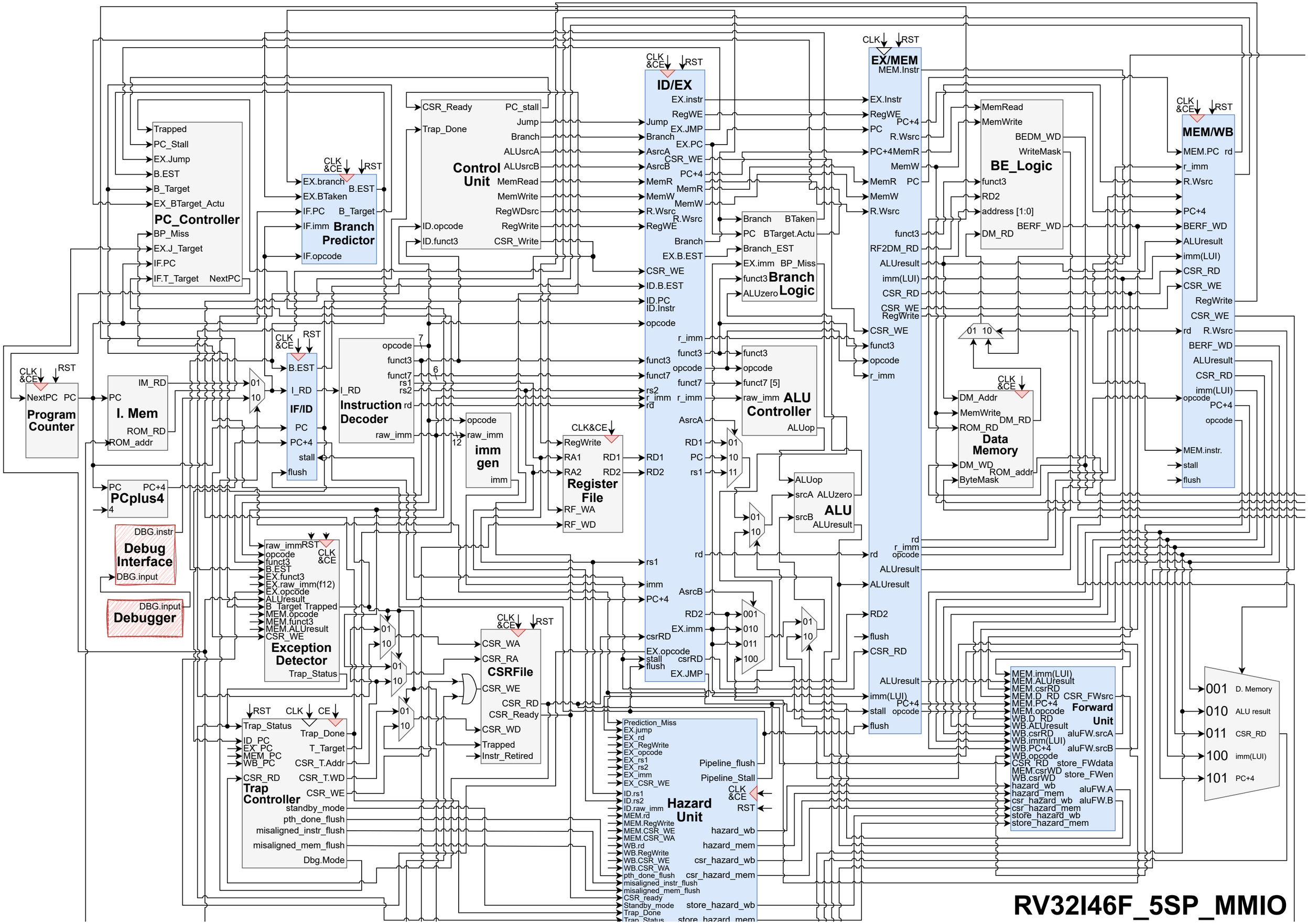
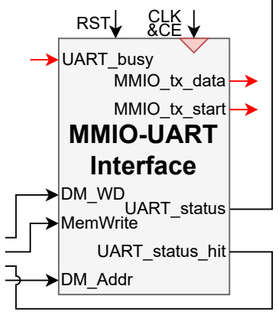
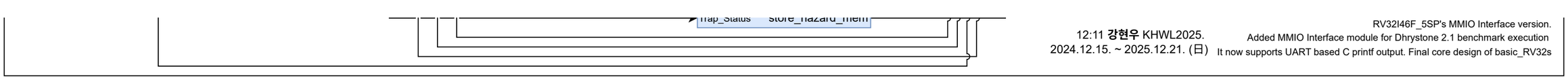

Program Counter
NextPC PC
CLK &CE
RST
I. Mem
IM_RD
PC
ROM_RD
ROM_addr
PCplus4
PC PC+4
4
Debug Interface
DBG.instr
DBG.input
Debugger
DBG.input
PC_Controller
Trapped
PC_Stall
EX.Jump
B.EST
B_Target
EX_BTarget_Actu
BP_Miss
EX.J_Target
IF.PC
IF.T_Target NextPC
Branch Predictor
EX.branch B.EST
EX.BTaken
IF.PC B_Target
IF.imm
IF.opcode
Control Unit
CSR_Ready PC_stall
Trap_Done Jump
Branch
ALUsrcA
ALUsrcB
MemRead
MemWrite
RegWDsrc
ID.opcode RegWrite
ID.funct3 CSR_Write
IF/ID
B.EST
I_RD
PC
PC+4
stall
flush
Instruction Decoder
I_RD
opcode
funct3
funct7
rs1
rs2
rd
raw_imm
7
6
12
imm gen
opcode
raw_imm
imm
Register File
CLK&CE
RegWrite
RA1 RD1
RA2 RD2
RF_WA
RF_WD
ID/EX
EX.instr
RegWE
Jump EX.JMP
Branch EX.PC
AsrcA CSR_WE
AsrcB PC+4
MemR MemR
MemW MemW
R.Wsrc R.Wsrc
RegWE
Branch
EX.B.EST
CSR_WE
ID.B.EST
ID.PC
ID.Instr
opcode
r_imm
funct3
funct7
rs2
r_imm
rd
AsrcA
RD1
RD2
rs1
imm
AsrcB
PC+4
RD2
EX.imm
csrRD
EX.opcode
stall csrRD
flush
EX.JMP
rd
Branch Logic
Branch BTaken
PC BTarget.Actu
Branch_EST
EX.imm BP_Miss
funct3
ALUzero
ALU Controller
funct3
opcode
funct7 [5]
raw_imm
ALUop
ALU
ALUop
srcA ALUzero
srcB
ALUresult
EX/MEM
MEM.Instr
EX.Instr
RegWE
PC PC+4
R.Wsrc
PC+4 MemR
MemW
MemR PC
MemW
R.Wsrc
funct3
RF2DM_RD
ALUresult
imm(LUI)
CSR_RD
CSR_WE RegWrite
CSR_WE
funct3
opcode
r_imm
rd
r_imm
opcode
ALUresult
ALUresult
RD2
flush
CSR_RD
ALUresult
imm(LUI)
stall PC+4 opcode
flush
MemRead
MemWrite
BEDM_WD
WriteMask
BE_Logic
funct3
RD2
address [1:0]
DM_RD BERF_WD
01 10
DM_Addr
MemWrite DM_RD
ROM_RD
Data Memory
DM_WD ROM_addr
ByteMask
MEM/WB
MEM.PC rd
r_imm
R.Wsrc
PC+4
BERF_WD
ALUresult
imm(LUI)
CSR_RD
CSR_WE
RegWrite
CSR_WE
rd R.Wsrc
BERF_WD
ALUresult
CSR_RD
imm(LUI)
opcode PC+4
opcode
MEM.instr.
stall
flush
001 D. Memory
010 ALU result
011 CSR_RD
100 imm(LUI)
101 PC+4
raw_imm
opcode
funct3
B.EST
EX.funct3
EX.raw_imm(f12)
EX.opcode
ALUresult
B_Target Trapped
MEM.opcode
MEM.funct3
MEM.ALUresult
CSR_WE
Exception Detector
Trap_Status
CSRFile
CSR_WA
CSR_RA
CSR_WE
CSR_RD
CSR_Ready
CSR_WD
Trapped
Instr_Retired
Trap_Status
ID_PC
EX_PC
MEM_PC
WB_PC
CSR_RD
Trap Controller
Trap_Done
T_Target
CSR_T.Addr
CSR_T.WD
CSR_WE
standby_mode
pth_done_flush
misaligned_instr_flush
misaligned_mem_flush
Dbg.Mode
Prediction_Miss
EX.jump
EX_rd
EX_RegWrite
EX_opcode
EX_rs1
EX_rs2
EX_imm
EX_CSR_WE
ID.rs1
ID.rs2
ID.raw_imm
MEM.rd
MEM.RegWrite
MEM.CSR_WE
MEM.CSR_WA
WB.rd
WB.RegWrite
WB.CSR_WE
WB.CSR_WA
pth_done_flush
misaligned_instr_flush
misaligned_mem_flush
CSR_ready
Standby_mode
Trap_Done
Trap_Status
Hazard Unit
Pipeline_flush
Pipeline_Stall
CLK &CE
RST
hazard_wb
hazard_mem
csr_hazard_wb
csr_hazard_mem
store_hazard_wb
store_hazard_mem
Forward Unit
MEM.imm(LUI)
MEM.ALUresult
MEM.csrRD
MEM.D_RD CSR_FWsrc
MEM.PC+4
MEM.opcode
WB.D_RD
WB.ALUresult
WB.csrRD aluFW.srcA
WB.imm(LUI)
WB.PC+4 aluFW.srcB
WB.opcode
CSR_RD store_FWdata
MEM.csrWD store_FWen
WB.csrWD
hazard_wb aluFW.A
hazard_mem
csr_hazard_wb aluFW.B
csr_hazard_mem
store_hazard_wb
store_hazard_mem
RV32I46F_5SP_MMIO


RST
CLK
&CE
UART_busy
MMIO_tx_data
MMIO_tx_start
MMIO-UART
Interface
DM_WD
UART_status
MemWrite
UART_status_hit
DM_Addr



12:11 **강현우** KHWL2025.
2024.12.15. ~ 2025.12.21. (日)

RV32I46F_5SP's MMIO Interface version.
Added MMIO Interface module for Dhrystone 2.1 benchmark execution
It now supports UART based C printf output. Final core design of basic_RV32s