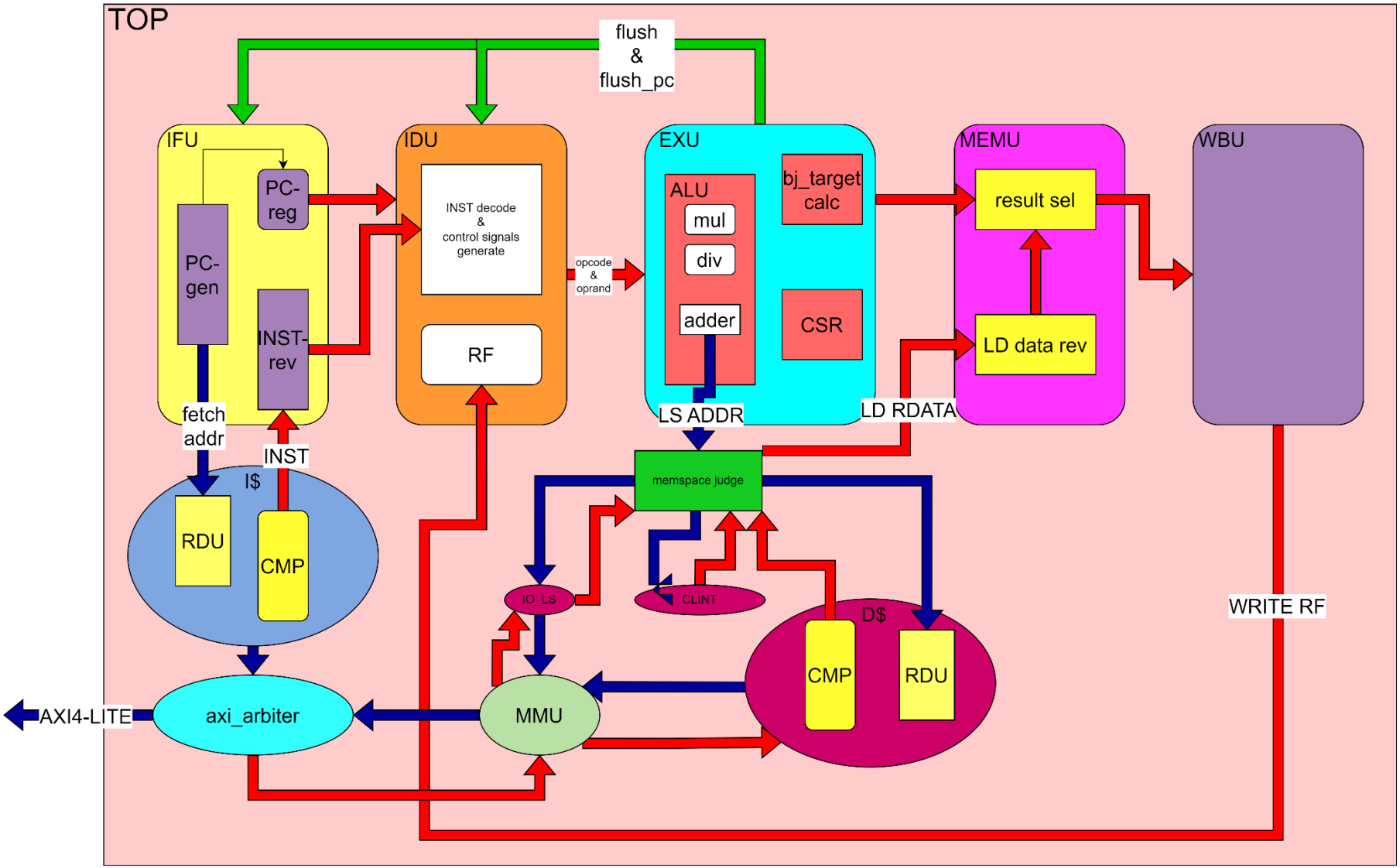
TOP
flush
&
flush_pc
IFU
PC-
reg
PC-
gen
INST-
rev
fetch
addr
INST
I$
RDU
CMP
IDU
INST decode
&
control signals
generate
RF
opcode
&
oprand
EXU
ALU
mul
div
adder
bj_target
calc
CSR
LS ADDR
LD RDATA
MEMU
result sel
LD data rev
WBU
WRITE RF
memspace judge
IO_LS
CLINT
D$
CMP
RDU
MMU
axi_arbiter
AXI4-LITE

IFU:取指单元

PC-gen 是一部分生成取指地址的组合逻辑，实现功能包括指令的预译码，静态分支预测，正常情况下的 PC+4 和 EXU 产生流水线刷新信号（flush）时的处理，给出最终确定的 next_pc。

PC-reg 寄存器存放已经发出取指请求等待返回指令的 pc。

INST-rev 直接将得到的指令内容与 PC-reg 一起送入下一流水级（译码）。

I$:指令 cache

RDU：发起对 cache 中 RAM 的读请求。

CMP：根据从 RAM 中读出的数据判断 cache 是否命中，若命中则直接返回所需指令；否则发起对 memory 的读请求，并返回 cache miss 信号以阻塞流水线，直到返回了所需指令。

Cache 支持流水化操作，即相邻的两条指令访问 cache 时，当前 1 条指令判断是否命中时的同时可以发起后一条指令的 RAM 读请求。

IDU：译码单元

RF：通用寄存器堆

INST decode & control signals generate：指令译码，产生后续指令执行所需个控制信号

EXU：执行单元

ALU：执行指令所需的算数逻辑运算，包括计算指令类的计算，访存类指令的访存地址计算，branch 指令的是否应该跳转的计算（复用了 ALU 中的加法器）。

Bj_target_calc：1 个加法器，计算 branch 指令和 jalr 指令的目标地址。

CSR：控制状态寄存器

Memspace judge：判断访存地址在内存地址空间对应的范围，在内存范围内则访问 D$，在 clint 范围内访问 mtime 寄存器，mtimecmp 寄存器，否则访问 IO 设备。

D$:数据 cache

与 icache 类似，不过要实现对 fence_i 指

令的支持，遍历所有 cache 块，将 dirty 位有效的块写回主存。

MMU:内存管理单元

对因为访问 IO 设备发起的内存访问请求和 dcache 发起的内存访问请求进行仲裁，同一时刻只能有 1 个请求被送入 AXI 仲裁器。

MEMU:访存流水级

接收执行级的运算结果和 Load 指令的数据，产生最终的指令执行结果送入写回流水级。

WBU:写回流水级

将指令的最终执行结果更新到寄存器堆。