1.2 Instruction Format for 16-bit ADSD-RISC ISA

	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
R-type	opcode	rs	rt	rd
S-type	opcode	rs	rt	#imm4
B-type	opcode	rs	rt	#imm4
L-type	opcode	rs	rt	#imm4
J-type	opcode	#imm12		

R-type : add, sub, or, and
S-type : shl, shr, rol, ror, not, addi
B-type : beq, blt, bgt
L-type : ld, st
J-type : jmp

1.3 Microarchitecture of ADSD-RISC

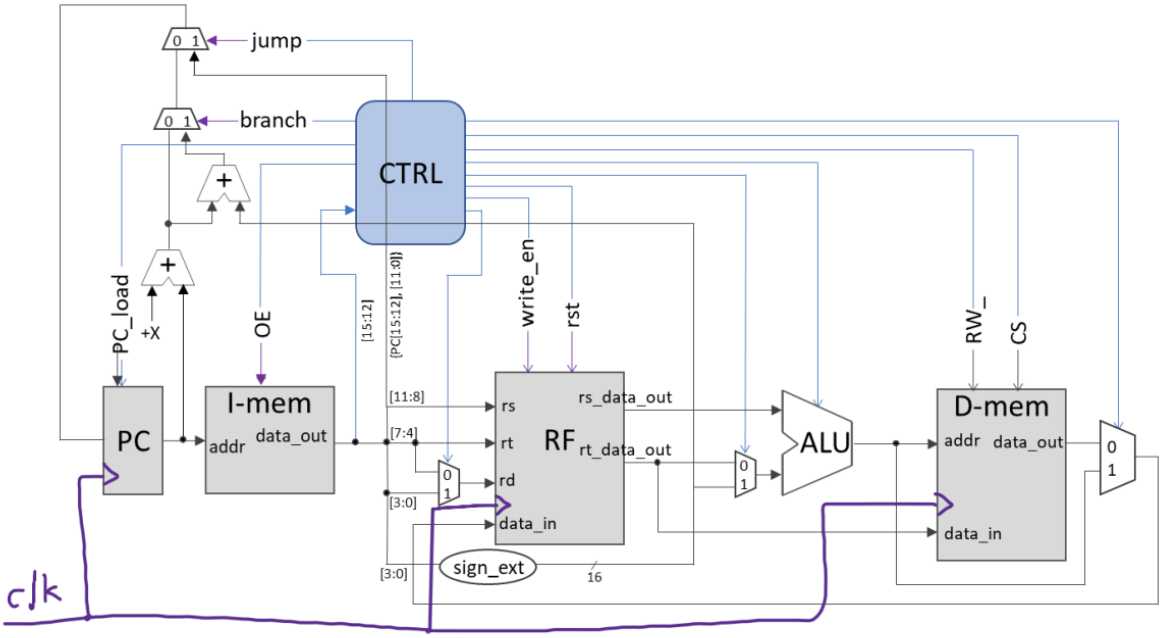

Figure 1.2 Block diagram of the single cycle microarchitecture of ADSD-RISC with the controller module.

2.0 Instruction Formats

2.1.1 R-type : Register instructions

Opcode	Instruction	RTL Operation
0000	add rs, rt, rd	rd ← rs + rt
0001	sub rs, rt, rd	rd ← rs + rt
0010	or rs, rt, rd	rd ← rs | rt (bitwise OR)
0011	and rs, rt, rd	rd ← rs & rt (bitwise AND)

2.1.2 S-type : Shift instructions

Opcode	Instruction	RTL Operation
0100	shl rs, rt, #imm4	rt ← arithmetic shift left rs by #imm4 bits
0101	shr rs, rt, #imm4	rt ← arithmetic shift right rs by #imm4 bits
0110	rol rs, rt	rt ← {rs[14:0], rs[15]}
0111	ror rs, rt	rt ← {rs[10], rs[15:1]}
1000	not rs, rt	rt ← ~rs //bitwise negation
1111	addi rs, rt, #imm4	rt ← rs + {12{imm4[3], imm4}

2.1.3 B-type : Conditional branch instructions

Opcode	Instruction	RTL Operation
1001	beq rs, rt, #imm4	If rs == rt, PC ← PC + 2 + #imm4, else PC ← PC + 2
1010	blt rs, rt, #imm4	If rs < rt, PC ← PC + 2 + #imm4, else PC ← PC + 2
1011	bgt rs, rt, #imm4	If rs > rt, PC ← PC + 2 + #imm4, else PC ← PC + 2

2.1.4 L-type : Load/store instructions

Opcode	Instruction	RTL Operation
1100	ld rs, rt, #imm4	rt ← DMEM [rs + #imm4]
1101	st rs, rt, #imm4	DMEM [rs + #imm4] ← rt

* #imm4 is sign-extended to 16 bits

2.1.5 J-type : Jump instruction

Opcode	Instruction	Operation
1110	jmp #imm	PC ← {PC[15:12], #imm12}