Data transfer (instruction formats)	R-I	R-I	R-I, R-R	R-I	R-I, R-R
Instruction name	**ARMv8**	**MIPS64**	**Power**	**RV64G**	**SPARC**
Load byte signed/unsigned.	LDR_B	LB_	LBZ; EXTSB	LB_	LD_B
Load halfword signed, unsigned	LDR_H	LH_	LHA/LHZ	LH_	LD_H
Load word	LDRSW/LDR	LW_	LW_	LW_	LD_W
Load double	LDRX	LD	LD	LD	LDD
Load float register SP/DP	LD_	L_C1	LF_	FL_	LD_F
Store byte	STB	SB	STB	SB	STB
Store half word	STW	SB	STB	SB	STB
Store word	STL	SW	STW	SW	ST
Store double word	STX	SD	SD	SD	STD
Store float SP/DP	ST_	S_C1	STF_	FS_	ST_F
Load reserved	LDEXB, LDEXH LDEXW, LDEXD	LL, LLD	lwarx,ldarx	LR	
Store conditional	STEXB, STEXH STEXW, STEXD	SC, SCD	stwcx, stdcx	SC	
Read/write spec. register	MF_, MT_	MF, MT_	M_SPR,	csrr_, csrr_i,	RD_, WR_
Move integer to FP register	ITOFS	MFC1/DMFC1	STW; LDFS	STW; FLDWX	ST; LDF
Move FP to integer register	FTTOIS	MTC1/DMTC1	STFS; LW	FSTWX; LDW	STF;LD
Synchronize data, instruction stream	DSB ISB	SYNC, SYNCI	SYNC, ISYNC	Fence Fence.i	MEMBAR FLUSH
Atomic operations	LDWAT, STWAT,	LDDAT STDAT	LLWP, LLDP, SCWP, SCDP	AMOSWAP .W/D, AMOADD, W.D, AMOAND.W/D, AMOOR.W/D, AMOMIN_.W/D, AMOMAX_W/D	CASA, SWAP, LDSTUB

FIGURE E.9 Desktop RISC data transfer instructions equivalent to RV64G core. A sequence of instructions to synthesize a RV64G instruction is shown separated by semicolons. Th e MIPS and Power instructions for atomic operations load and conditionally store a pair of registers and can be used to implement the RV64G atomic operations with at most one intervening ALU instruction. The SPARC instructions: compare-and-swap, swap, LDSTUB provide atomic updates to a memory location and can be used to build the RV64G instructions. The Power3 instructions provide all the functionality, as the RV64G instructions, depending on a function field.