(d) Uarch and Dataflow of Instructions (a) Comparison Co-processor WBTILE Vrf rdata (b) Common Impletation Custom Mem access addr Instruction Mem rdata 1 inst enable ctrl unit Mem for (i = 0; i < 4; i++)Access Funct7 Rs2 Rs1 Decoder Rd Opcode ct3 4x4 BTd[i] = d[0 + i] - d[8 + i];Vrf higher mem acc BTd[4+i] = d[4+i] + d[8+i];ifmap Write Data registers priority waddr BTd[8+i] = -d[4+i] + d[8+i];Imple LDTIL port conv **Issue logic** BTd[12 + i] = d[4 + i] - d[12 + i];. ment Vrf raddrs **PPVCU** frequeenv † **DATA** 2x2Valid in ofmap for (i = 0; i < 4; i++)**CONTAINER** pipelined next V[0 + i * 4] = BTd[0 + i * 4] - BTd[2 + i * 4];**STAGE1-1Cycle** STAGE2-1Cycle V[1+i*4] = BTd[1+i*4] + BTd[2+i*4];V[2+i*4] = -BTd[1+i*4] + BTd[2+i*4];(c) Custom Impletation V[3 + i * 4] = BTd[1 + i * 4] - BTd[3 + i * 4];to stage3 res1 vec src0 ADDER Vrf **ADDER** LDTILE &U,16,vec{7-10}; vec src1 **AAMUL** vec src2 For (i=0;i<16;i++)rdatas write invariant: UV[i] = U[i] * V[i];**Prefetch** LDTILE &d,16,vec{0-3}; Inst infomation **Inst infomation** Crtl AAMUL vec7,vec2,vec0,vec0; TRIADD LDTILE &d+16,16,vec{tmps}AMUL vec8,vec2,vec1,vec4; for (int i = 0; i < 4; i++) OACC ATUV[i] = UV[0+i] + UV[4+i] + UV[8+i];STAGE3-2Cycles AAMUL vec9, vec1, vec2, vec5; ATUV[4 + i] = UV[4 + i] - UV[8 + i] - UV[12 + i];AAMUL vec10, vec3, vec1, vec3; PIPELINED TRIADD vec4, vec5, vec0, vec0; write Arbiter Y[0] = (ATUV[0] + ATUV[1] + ATUV[2]);**VECTOR** MULTIPLIER vec4,vec5,vec3,vec3; From vec src2 Y[1] = (ATUV[1] - ATUV[2] - ATUV[3]);vec0, vec6, vec6; OACC COMPUTING stage2 Vrf Y[2] = (ATUV[4] + ATUV[5] + ATUV[6]);wmask waddr vec3, vec6, vec6; lower **UINT** Write Y[3] = (ATUV[5] - ATUV[6] - ATUV[7]);priority WBTILE vec6,&Y; Port