5 Connections to Classic Architectures

5.a IBM System/360 Model 91 with the Tomasulo Algorithm

The IBM System/360 Model 91 supports out of order writeback/completion as shown by the example instruction sequence on pg.171 of Hennessy and Patterson. Register renaming was added to eliminate WAW and WAR hazards. Furthermore, System/360 also has out of order issue (in the textbook, this is stated as the execute stage; pg. 174 of Hennessy and Patterson). Instructions are issued in order from the issue queue, but stalled in the functional units until their source operands are available. This is functionally the same as an out of order issue stage that we have in the IO2E and IO2L microarchitectures. Finally, the writeback stage is most similar to a late commit system. As stated on pg. 172, if multiple instructions write to the same destination register, only the latest instruction is used to update the register file. Therefore, the IBM System/360 Model 91 is most similar to an IO2L architecture.

5.b Register Renaming in the MIPS R10K and the Intel P6 Microarchitectures

The MIP R10000 microarchitecture uses a pointer-based register renaming scheme. As stated on page 33 of the report (see source), the register mapping system contains register map tables, free lists, active list, and busy-bit tables. The register map tables takes on the duty of the rename table in our scheme. The register map table keeps track of all current mappings from the PRF to the ARF. The active list then acts as the ROB, keeping track of all current instructions that are active in the processor. The active list holds the ARF register and the previous PRF register. The free list acts exactly like the FL in our pointer-based scheme. The busy-bit tables simply act as the list of valid bits in the ROB table. When combined, these units function as a pointer-based register renaming system.
Source: The MIPS R10000 Superscalar Microprocessor by Kenneth C. Yeager
(http://people.cs.pitt.edu/ cho/cs2410/papers/yeager-micromag96.pdf)

The Intel P-6 microarchitecture uses a register renaming scheme that is similar to a value-based scheme. As stated on page 2 of the tech report (see source), the RAT (register alias table), which is similar to our rename table, determines whether an operand should be taken from the RRF (real register file), which is similar to our ARF, or from the ROB. If a source operand is currently pending, then the reservation station (similar to our issue queue) is given a pointer to a location in ROB instead and the instruction waits until that source operand is no longer pending. This means that all values are stored in either the ROB or the RRF, which is similar to our value-based scheme, where all values are stored in either the ROB or ARF.
Source: Intel's P6 Uses Decoupled Superscalar Design by Linley Gwennap
(http://www.cs.cmu.edu/afs/cs/academic/class/15213-f01/docs/mpr-p6.pdf)