Tomasulo's Algorithm

```
For each ALU
    If the instruction is complete
        If there is a RAT entry for the completed instruction
            Put the instruction result into the register that corresponds to
            that RAT entry.
            Mark the RAT entry as unused.
        For each RS
            For each operand slot in the RS
                If the operand slot is waiting for the results of the completed
                instruction
                    Put the result of the completed instruction into the operand
                    slot.
        Mark the ALU as unused.
        Mark as unused the RS that was occupied by the completed instruction.

While there is an instruction to issue
    If there is an appropriate RS entry which was empty at the beginning of the
    cycle, or which became empty during the cycle and the processor allows reuse
    in the same cycle
        Put the instruction into that RS entry.
        For each operand
            If the RAT entry corresponding to the operand register indicates
            that a result is being waited on
                Put the RS number of the instruction that we are waiting on into
                the operand slot of the RS for the instruction that is being issued.
            else
                Fetch the value from the operand register and put it into the
                operand of the RS for the instruction that is being issued.
        Put the RS number for the instruction that is being issued into the RAT
        entry that corresponds to the destination register of the instruction
        that is being issued.
        Remove the instruction from the instruction queue.

For each RS
    If RS has instruction with actual values for operands
        If the appropriate ALU or processing unit is free
            Dispatch the instruction, including operands and the RS number.
```

Reorder Buffer

```
For each ALU
    If the instruction is complete
        Put the result into the ROB entry corresponding to the destination register.
        For each RS that is waiting for it
            Put the result into the RS as an operand.
        Free the ALU.

While there is an instruction to issue
    If there is an empty ROB entry and an empty appropriate RS
        Put opcode into RS.
        Put ROB entry number into RS.
        For each operand which is a register
```

```
            If there is a ROB entry number in the RAT for that register
                Put the ROB entry number into the RS as an operand.
            else
                Put the register value into the RS as an operand.
        Put opcode into ROB entry.
        Put destination register name into ROB entry.
        Put ROB entry number into RAT entry for the destination register.
        Take the instruction out of the instruction window.

For each RS
    If RS has instruction with actual values for operands
        If an appropriate ALU or processing unit is free
            Dispatch the instruction, including operands and the ROB entry number.
            Free the RS.

While the next ROB entry to be retired has a result
    Write the result to the register.
    If the ROB entry number is in the RAT, remove it.
    Free the ROB entry.
```