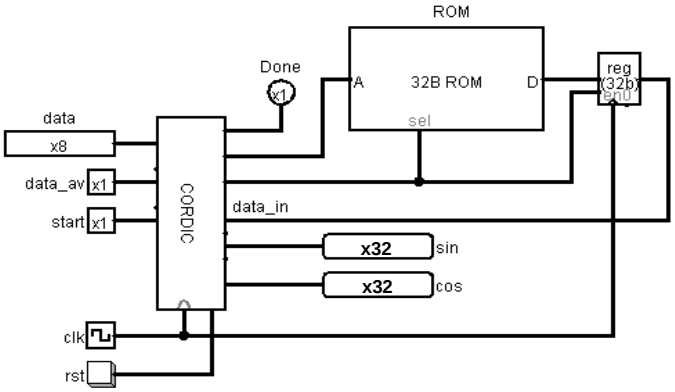
Trabalho 1 – parte 1

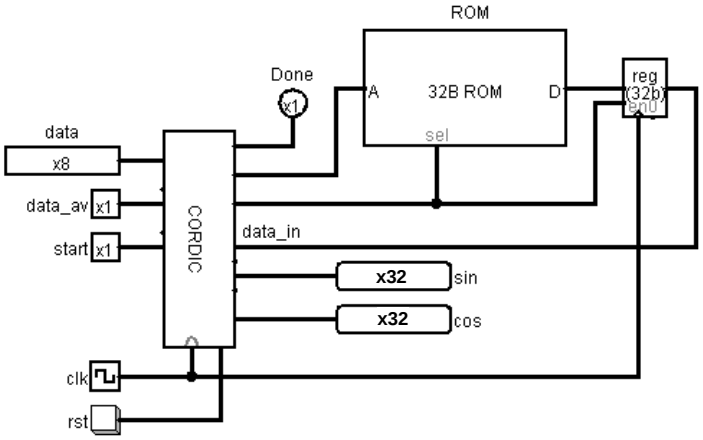
- cordic(int angle, int it, int *sin, int *cos)
  - Projetar um processador para calcular simultaneamente o seno e o cosseno de um ângulo através do algoritmo CORDIC
    - O ângulo e o número de iterações a serem executadas devem ser especificados na entrada *data* nesta ordem
    - A entrada *data_av* indica que a entrada *data* é valida
      - Para cada valor válido, deve ficar ativa por um ciclo de *clock*
    - Quando *Start* = 1, o processador começa o ordenamento
    - O fim do cálculo deve ser indicado ativando a saída *Done* por 1 ciclo de *clock*
    - As saídas *sin* e *cos* apresentam os valores calculados



Trabalho 1 – parte 1

- cordic(int angle, int it, int *sin, int *cos)
  - Os valores válidos de ângulos são de 0 a 90 graus (*int angle*)
  - O número máximo de iterações é 32 (*int it*)
  - O *datapath* deve operar com dados de 32 bits
    - Os dados de entrada devem ser expandidos para 32 bits internamente
  - Devido ao fato do algoritmo utilizar aritmética de inteiros para operar com números reais, o resultado calculado nas saídas *sin* e *cos* está deslocado 24 bits para a esquerda (multiplicado por 2^{24}) e deve ser interpretado como um número com a vírgula entre os bits 24 e 23 (ponto fixo)
    - A implementação C fornecida foi implementada desta maneira (*moodle*)
    - Ler documentação no *moodle* sobre o algoritmo (CORIDC for Dummies)

Os valores calculados devem ser idênticos aos fornecidos pela implementação C



Trabalho 1 – parte 1

- cordic(int angle, int it, int *sin, int *cos)
  - A memória utilizada é do tipo ROM e deve ser preenchida manualmente com os valores da tabela de ângulos da implementação C (*anglesTable*)
  - Foi adicionado um registrador na saída da dados da memória a fim de emular uma memória com leitura síncrona
  - Restrição de projeto: deve-se utilizar um único somador para realizar todas operações aritméticas e comparações