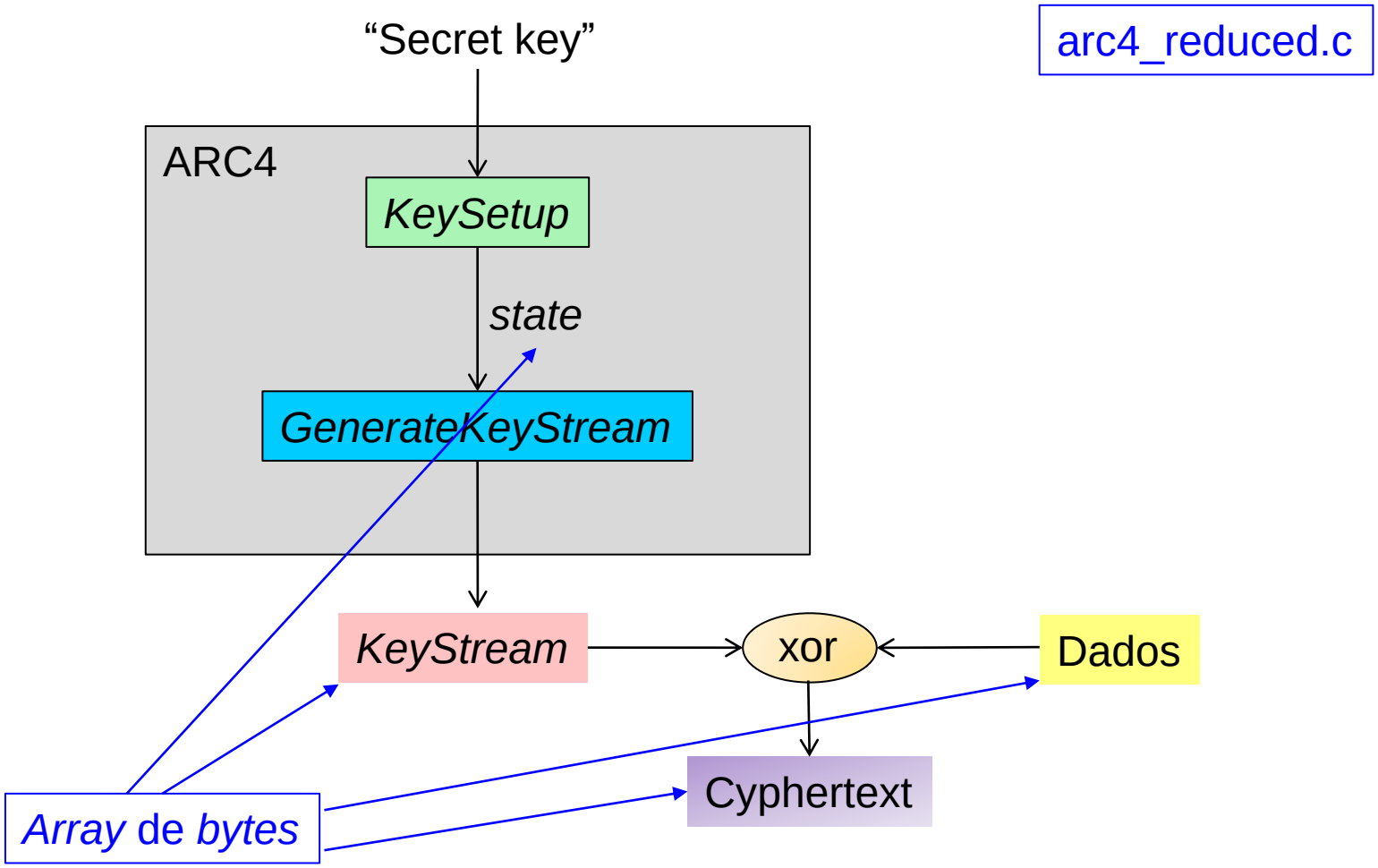
Trabalho – parte 1

- ARC4
  - Algoritmo utilizado em criptografia
  - Gera chave criptográfica utilizada na encriptação e decriptação de dados
  - Caiu em desuso, mas já fez parte de protocolos bastante conhecidos
    - WEP e WPA
      - Protocolos de segurança de rede sem fio que protegem dados através de criptografia. Se os dados sem fio forem interceptados, eles serão irreconhecíveis para os interceptores, uma vez que estão criptografados.
    - SSL e TCL
      - Protocolos de segurança digital que permitem a comunicação criptografada entre um navegador e um site
      - Um site que implementa esses protocolos tem HTTPS ao invés de HTTP no endereços

Trabalho – parte 1

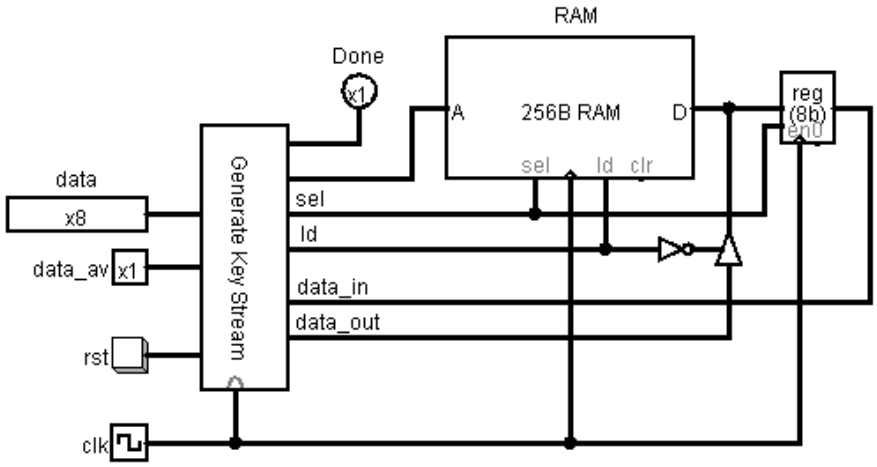
- ARC4



Trabalho – parte 1

- Projetar um processador que implemente a função GenerateKeyStream() do algoritmo ARC4
  - A implementação deve seguir o programa arc4_reduced.c disponível no *moodle*
  - O parâmetros *state*, *stateSize*, *textSize* e *keyStream* devem ser fornecidos através da entrada *data*
  - A entrada *data_av* indica que a entrada *data* é valida
    - Para cada parâmetro válido, deve ficar ativa por um ciclo de *clock*
  - Após ler o último parâmetro, o processador começa o processamento
  - Ao final, a saída *Done* deve ficar ativa por 1 ciclo de *clock*

O *keyStream* gerado deve ser idêntico ao gerado pela implementação C fornecida

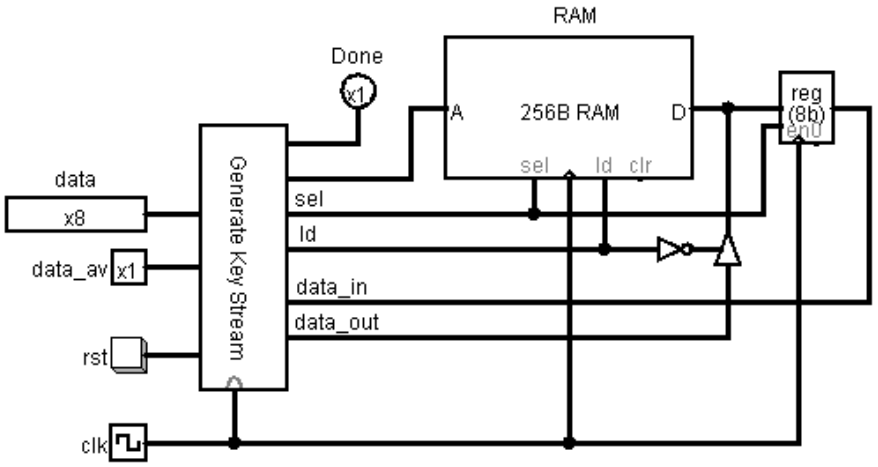


Trabalho – parte 1

- Projetar um processador que implemente a função GenerateKeyStream() do algoritmo ARC4
  - A implementação deve seguir o programa arc4_reduced.c disponível no *moodle*
  - O parâmetros *state*, *stateSize, textSize* e *keyStream* devem ser fornecidos através da entrada *data*

O conteúdo do *array state* é gerado pela função *KeySetup*(). Como esta função não será implementada, o *array* deve ser inserido manualmente em algum lugar na memória, de acordo com o código C

Deve ser fornecido para o processador o endereço inicial deste *array* na memória

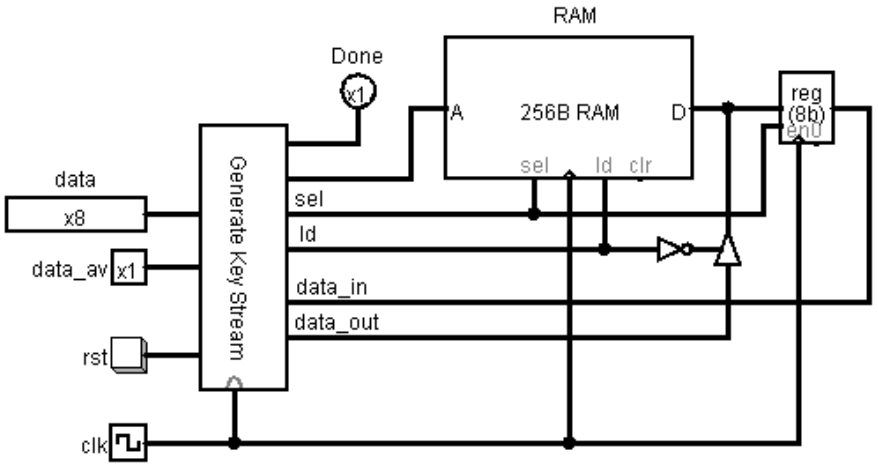


Trabalho – parte 1

- Projetar um processador que implemente a função GenerateKeyStream() do algoritmo ARC4
  - A implementação deve seguir o programa arc4_reduced.c disponível no *moodle*
  - O parâmetros *state*, *stateSize, textSize* e *keyStream* devem ser fornecidos através da entrada *data*

O *array keyStream* deve ser preenchido pelo processador, de acordo com o código C

Deve ser fornecido para o processador o endereço inicial deste *array* na memória

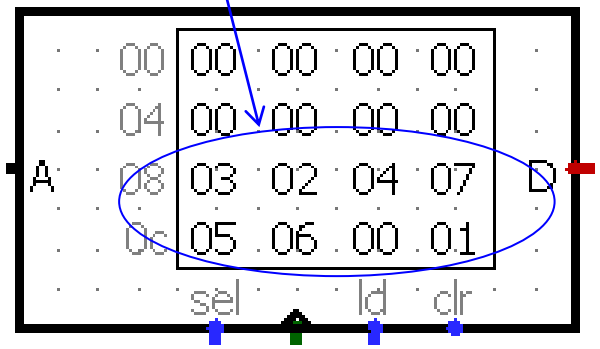


Trabalho – parte 1

- Exemplo
  - *state*: 0x08 (endereço inicial do *array*)
  - *stateSize*: 0x08
  - textSize: 5 (“teste”)
  - *keyStream*: 0x00 (endereço inicial do *array*)

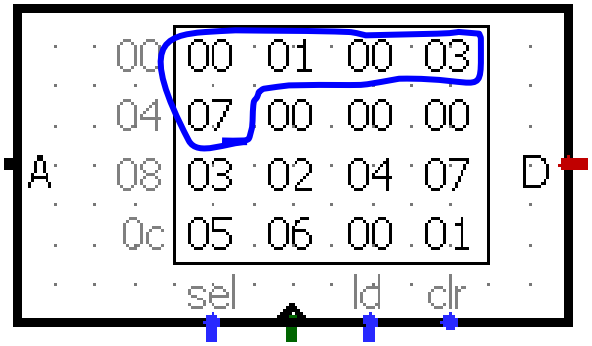
state = [0x03, 0x02, 0x04, 0x07, 0x05, 0x06, 0x00, 0x01]

Preenchido manualmente



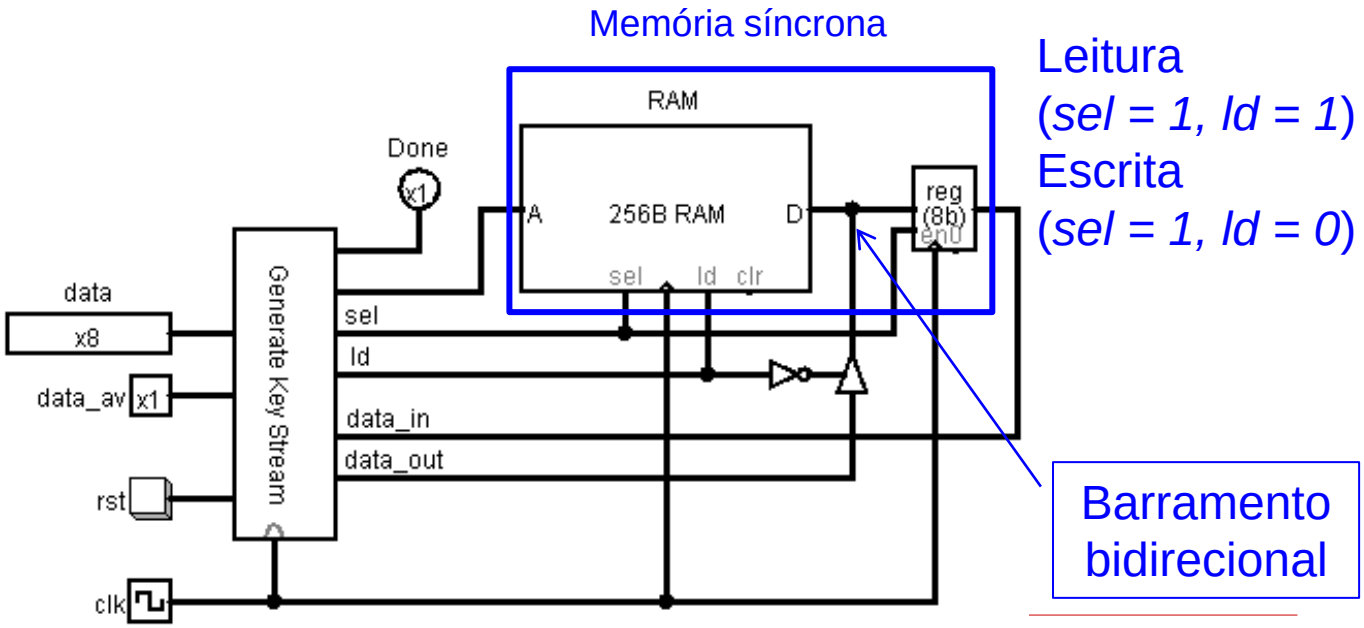
keyStream = [0x00, 0x01, 0x00, 0x03, 0x07, 0x00, 0x00, 0x00]

Preenchido pelo processador



Trabalho – parte 1

- Memória com leitura síncrona
  - Foi adicionado um registrador na saída da dados da memória a fim de emular uma memória com leitura síncrona
  - Restrição de projeto: NÃO DESPERDIÇAR HARDWARE! Replicar hardware com foco no desempenho



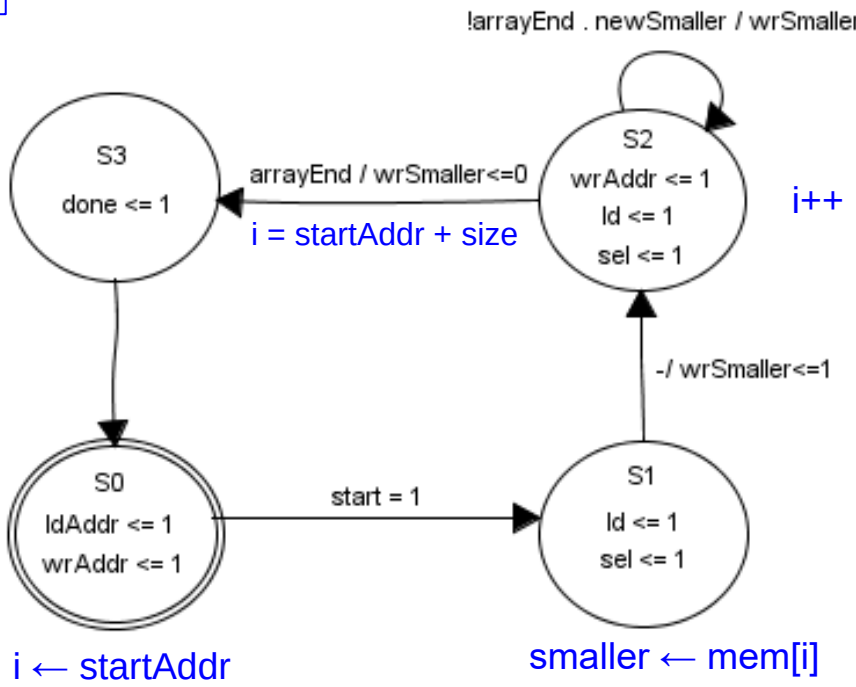
Trabalho – parte 1

- Trabalho a ser feito em trios
- Entrega dia 30/4 (todos trios)
  - Arquivo do logisim via *moodle*
    - Um integrante fica responsável pela submissão (.zip)
  - Bloco operativo (**impresso**)
    - Diagrama **claro e legível (sem teias de aranha!)**
    - Utilizem túneis para o *clock* e *reset* no *datapath* a fim de deixar o diagrama claro
    - Não utilizar túneis para os barramentos de dados
  - Grafo da FSM (**impresso**)
    - Sugestão de softwares
      - http://www.fizzim.com
      - https://www.yworks.com/products/yed
    - Atenção às notações Mealy/Moore e condições de transição
      - Apresentar somente os sinais relevantes em cada estado
    - Indicar em cada estado (fora do estado) a ação realizada pelo *datapath* (FSM + FSMD)

Trabalho – parte 1

- Exemplo de diagrama FSM + FSMD

Sinais sem valor explícito são iguais a '0'



FSM + FSMD

Trabalho – parte 1

- Apresentações dias 30/4 e 2/4
  - Cada grupo terá em torno de 15 minutos para apresentar junto ao professor
- Para a parte 2 do trabalho, os mesmos grupos deverão ser mantidos
  - Só estarão habilitados a realizar a parte 2 do trabalho os grupos que apresentarem a parte 1, mesmo que fora do prazo