Интерпретатор Brainfuck

ШУСТИКОВ В.Н.
ЕНИКЕЕВ Д.В.
ГР. 8306

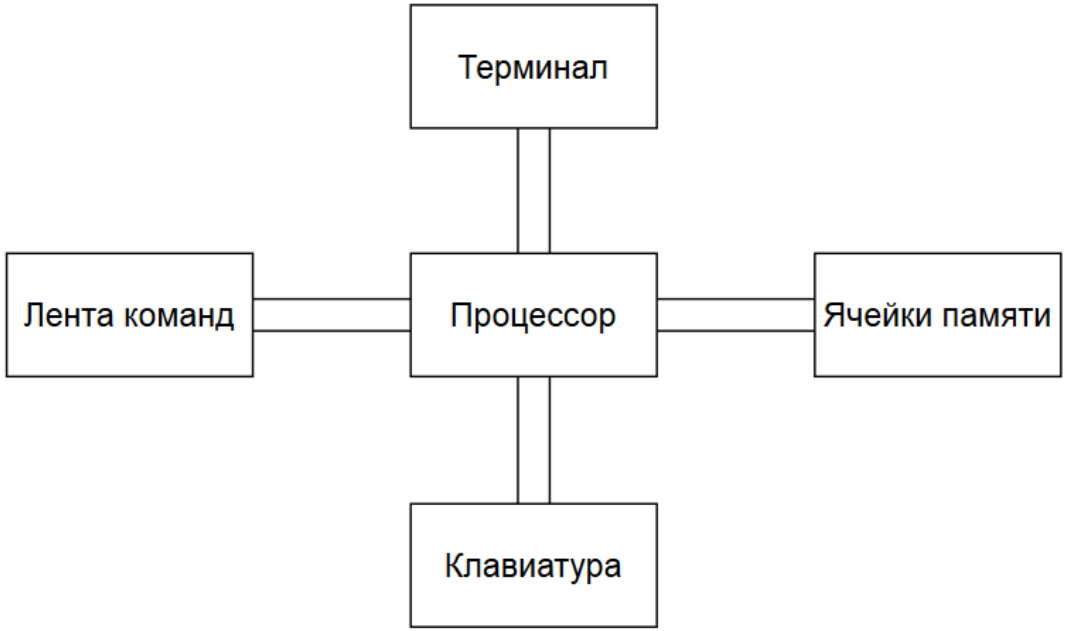
Язык программирования Brainfuck

- Минималистичный язык
- 8 команд
- Ввод с клавиатуры, вывод в терминал
- Бесконечное количество 8-бит ячеек памяти
- Полный по Тьюрингу

Цели

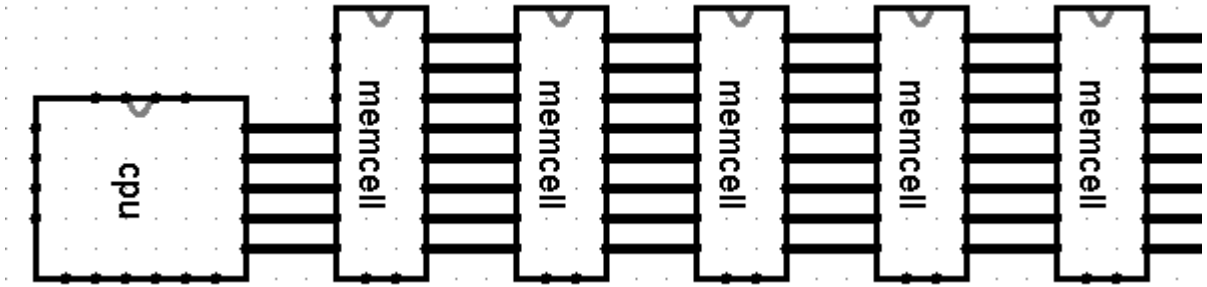
- Изучить возможность построения модели интерпретатора языка Brainfuck
- Спроектировать аппаратную реализацию интерпретатора Brainfuck с расширяемой памятью
- Реализовать интерпретатор языка Brainfuck на элементной базе симулятора Logisim

Описание работы модели

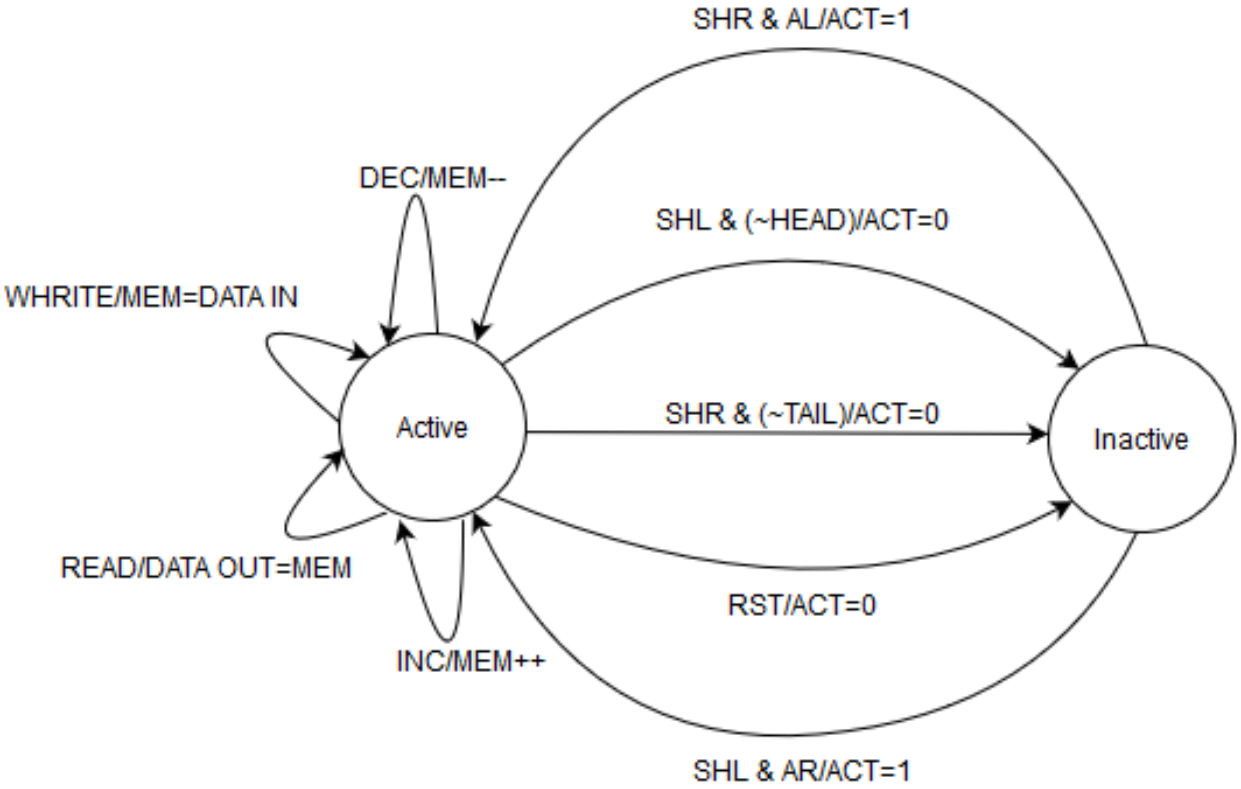


Ячейка памяти

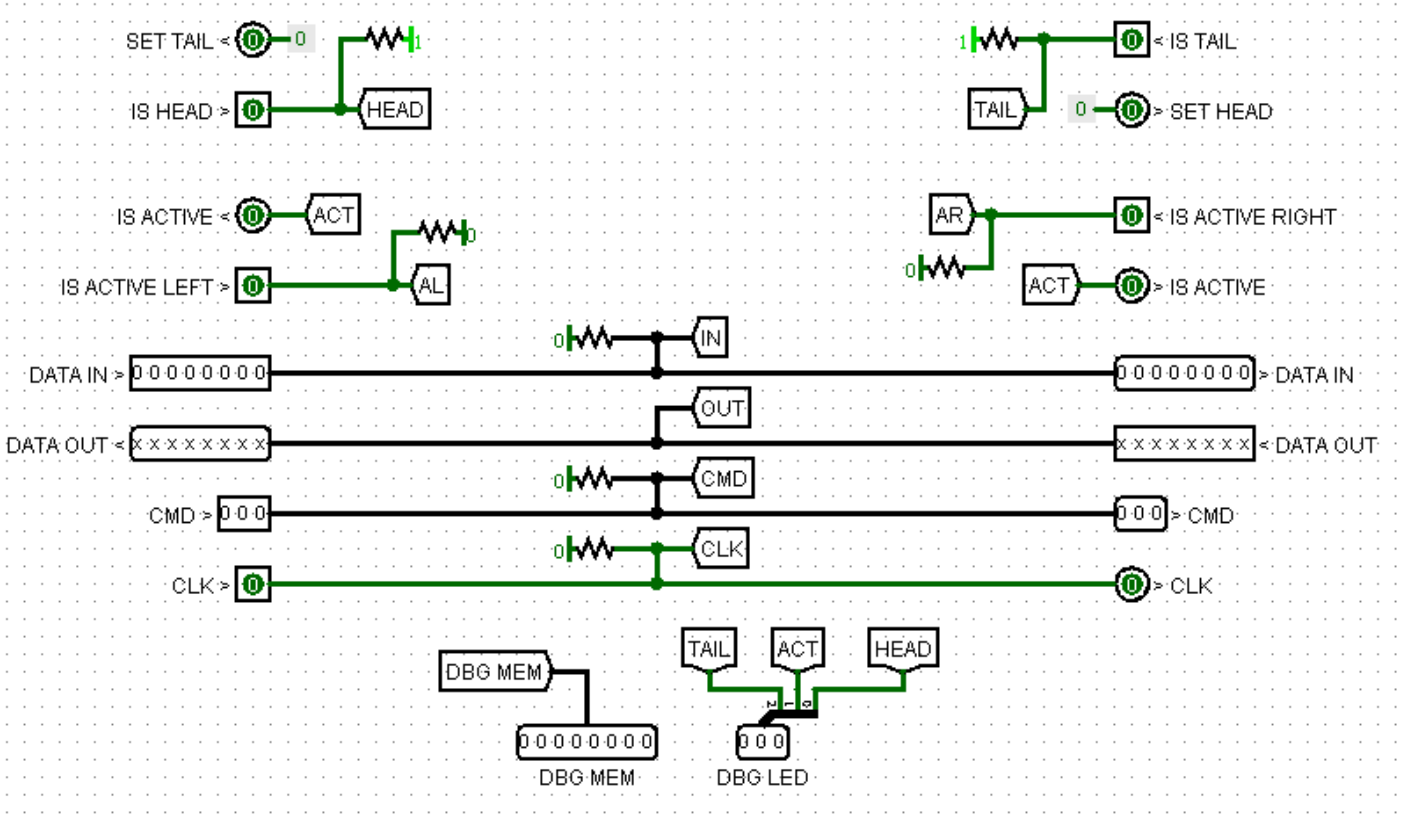
- Хранит 8 бит памяти и флаг активности
- Сквозная шина, последовательное подключение
- Видит окружение, чтобы переключать активность



Автомат ячейки памяти



Выходы микросхемы ячейки



Входные сигналы автомата

- CMD0 – младший бит входа CMD
- SHIFT – на входе CMD команда сдвига
- RW – на входе CMD команда чтения или записи
- ARITH – на входе CMD команда инкремента или декремента
- ZERO – на входе CMD команда сброса (если CMD0 = 1)
- AR – справа активная ячейка
- AL – слева активная ячейка
- H – ячейка является головой цепочки (подключена к CPU)
- T – ячейка является хвостом цепочки
- Q – текущее состояние активности

Выходные сигналы автомата

- COUNT и LOAD – сигналы на вход элемента-счетчика Logisim
- WRITE – открывает ключ для записи значения в счетчик
- OUT – открывает ключ для вывода значения из счетчика в шину
- ACT – следующее состояние активности

Формулы выходных сигналов

$$WRITE = CMD0 \wedge RW \wedge Q0$$
$$OUT = \overline{CMD0} \wedge RW \wedge Q0$$
$$LOAD = (CMD0 \wedge ARITH \wedge Q0) \vee (CMD0 \wedge RW \wedge Q0)$$
$$COUNT = ARITH \wedge Q0$$

Функция возбуждения элементов памяти

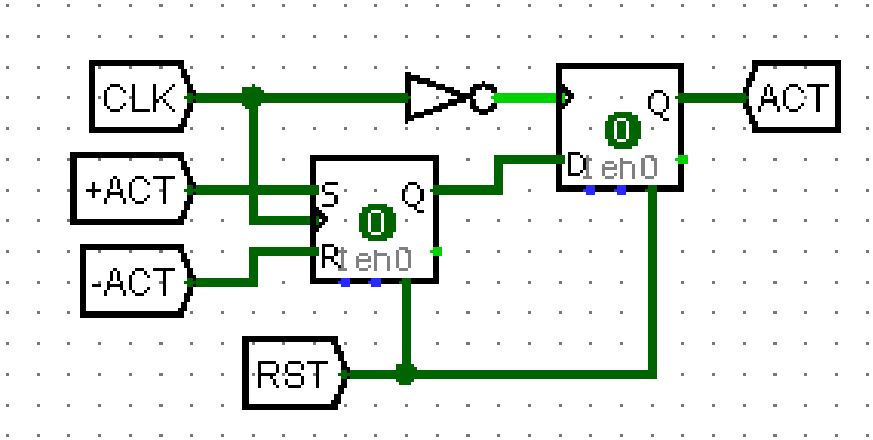
- При SHIFT=1:

$$D = (\mathrm{CMD0} \wedge AL \wedge \overline{Q}) \vee (\overline{\mathrm{CMD0}} \wedge AR \wedge \overline{Q}) \vee$$
$$\vee (\overline{\mathrm{CMD0}} \wedge \overline{H} \wedge Q) \vee (\mathrm{CMD0} \wedge T \wedge Q)$$

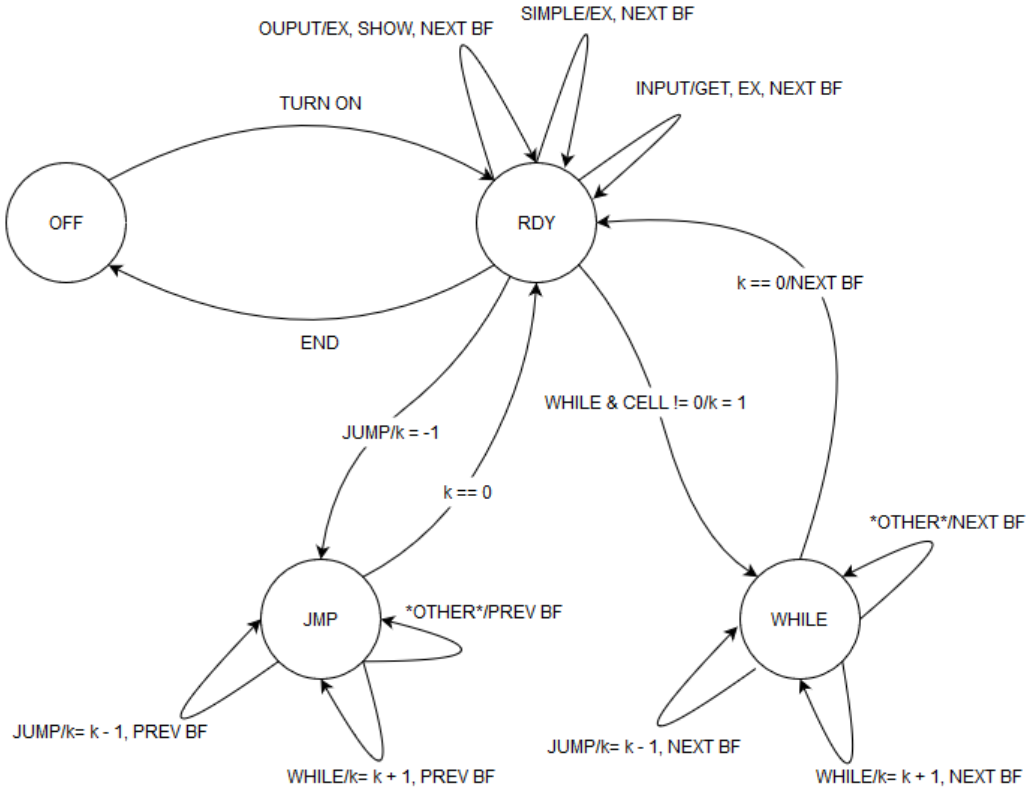
- При ZERO=1 и CMD0=1 на RST-вход счетчика и триггеров подается 1
- В остальных случаях D не меняется

Особенности смены активности

- Новая активность ячейки D зависит от текущей активности ее соседей AL и AR
- MS-триггер: задержка смены состояния

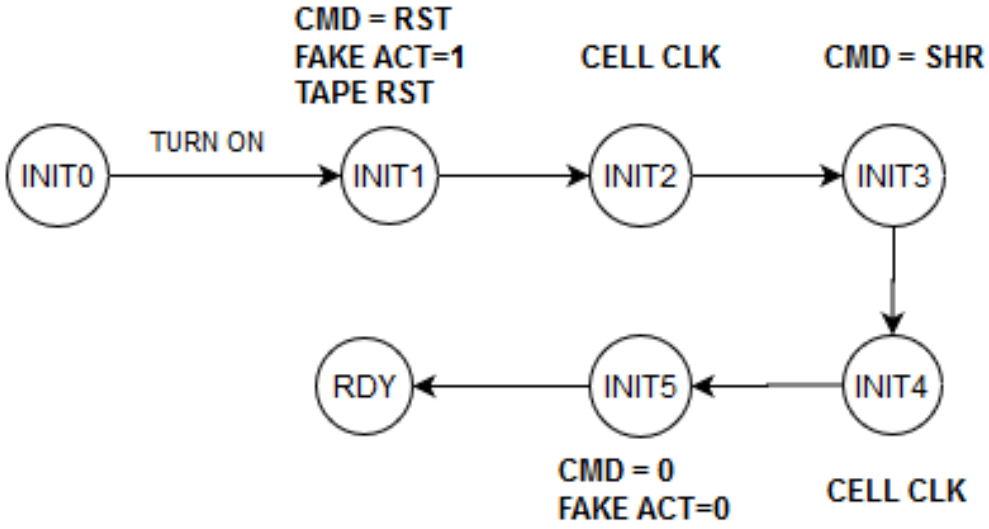


Общая схема процессора



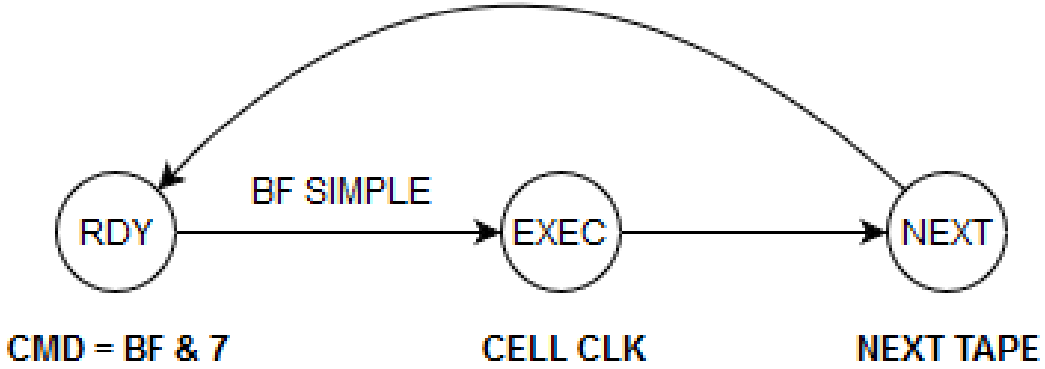
Инициализация

- Сброс ячеек памяти и сдвиг вправо



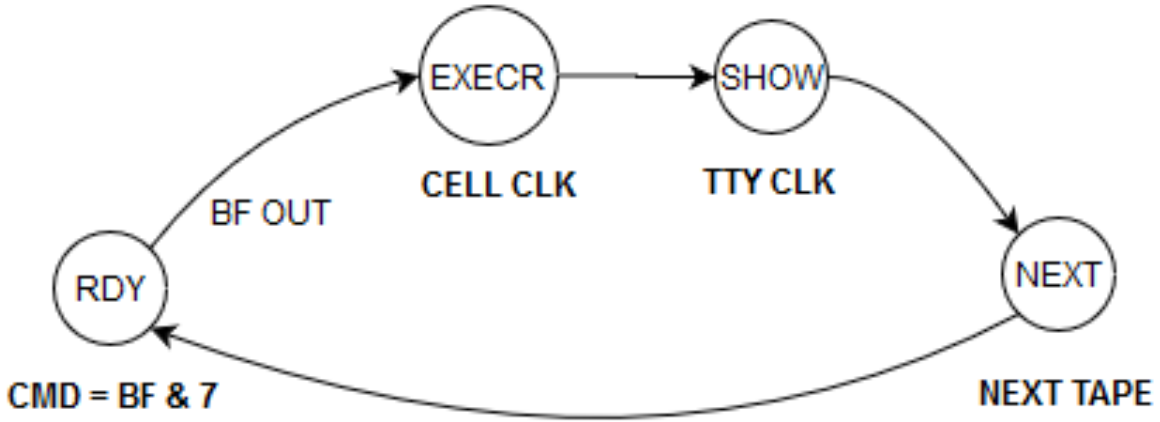
Простые команды Brainfuck

- Инкремент, декремент, сдвиги – выполняются ячейкой



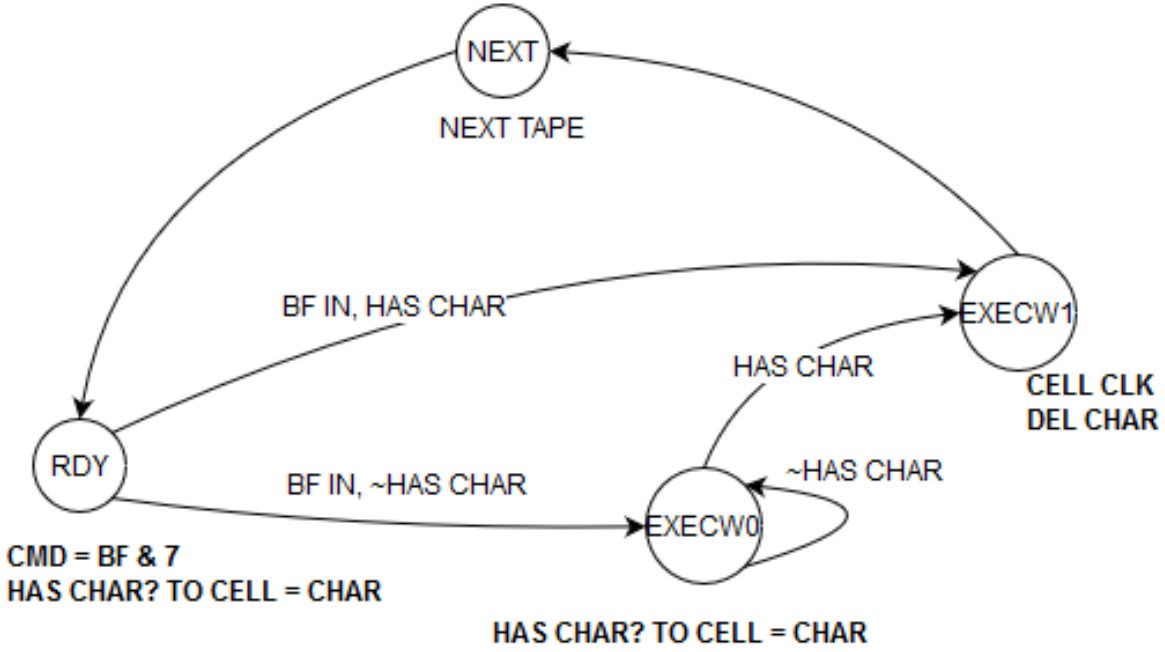
Вывод в терминал

- Посылка тактового импульса в терминал после чтения из ячейки



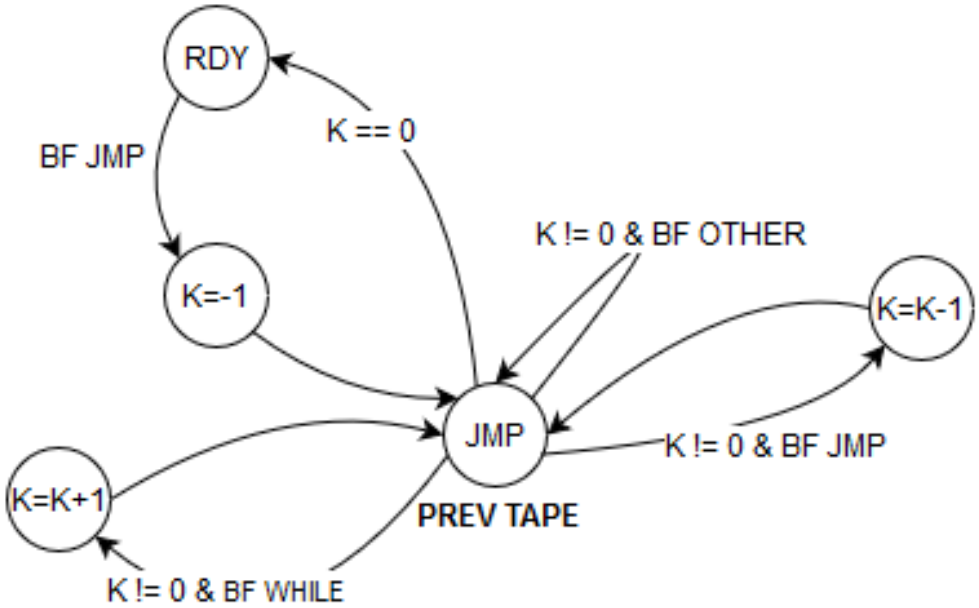
Ввод с клавиатуры

- Ожидание символа если буфер пуст

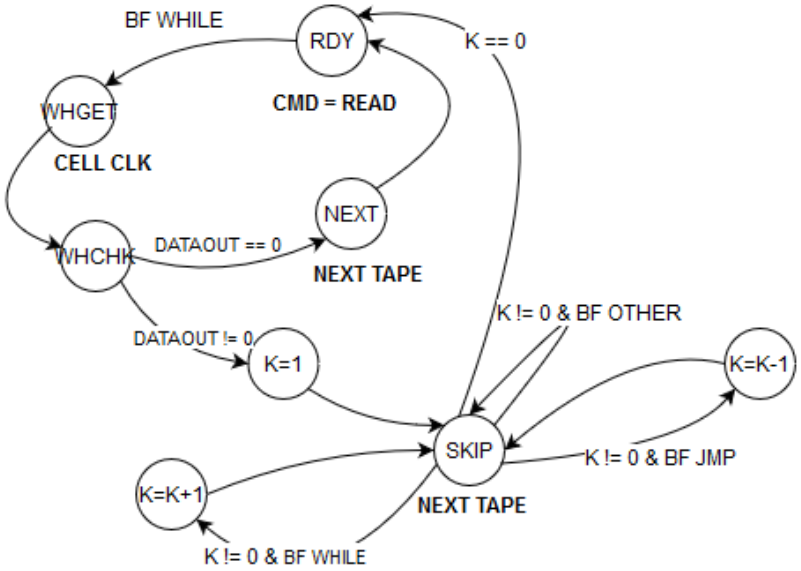


Команда JUMP

- Проход назад по ленте до парной команды WHILE



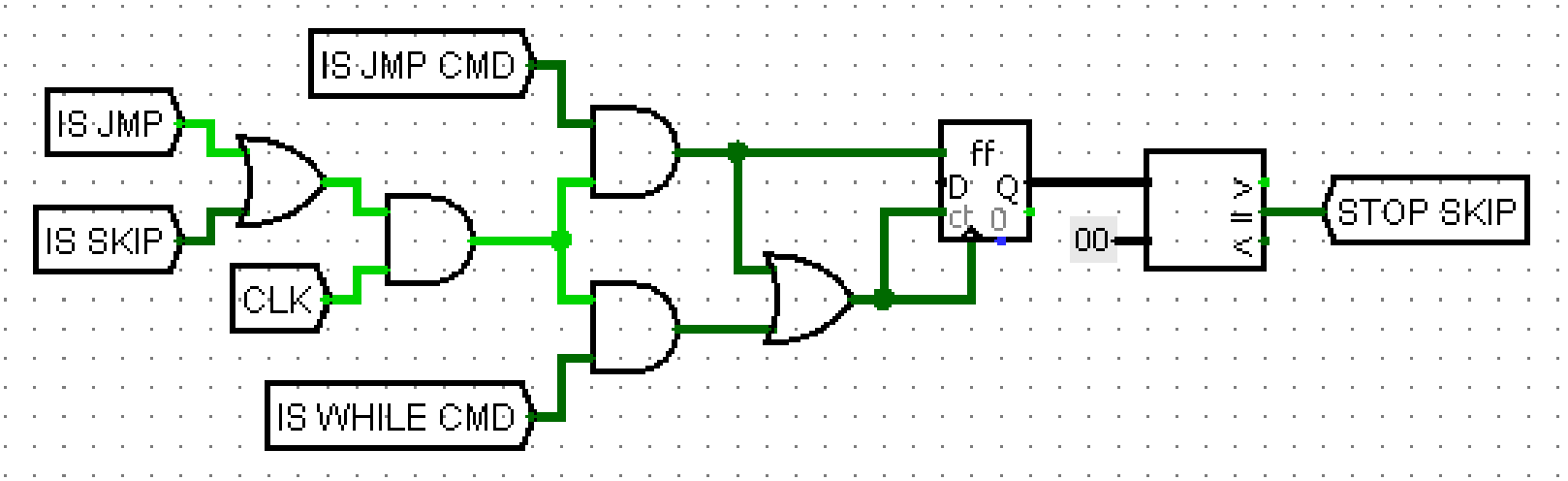
Команда WHILE



- Чтение значения из текущей ячейки и проверка на 0
- Проход вперед по ленте за парную команду JUMP

Счетчик баланса JUMP/WHILE

- Счетчик Logisim, который считает встреченные команды JUMP и WHILE

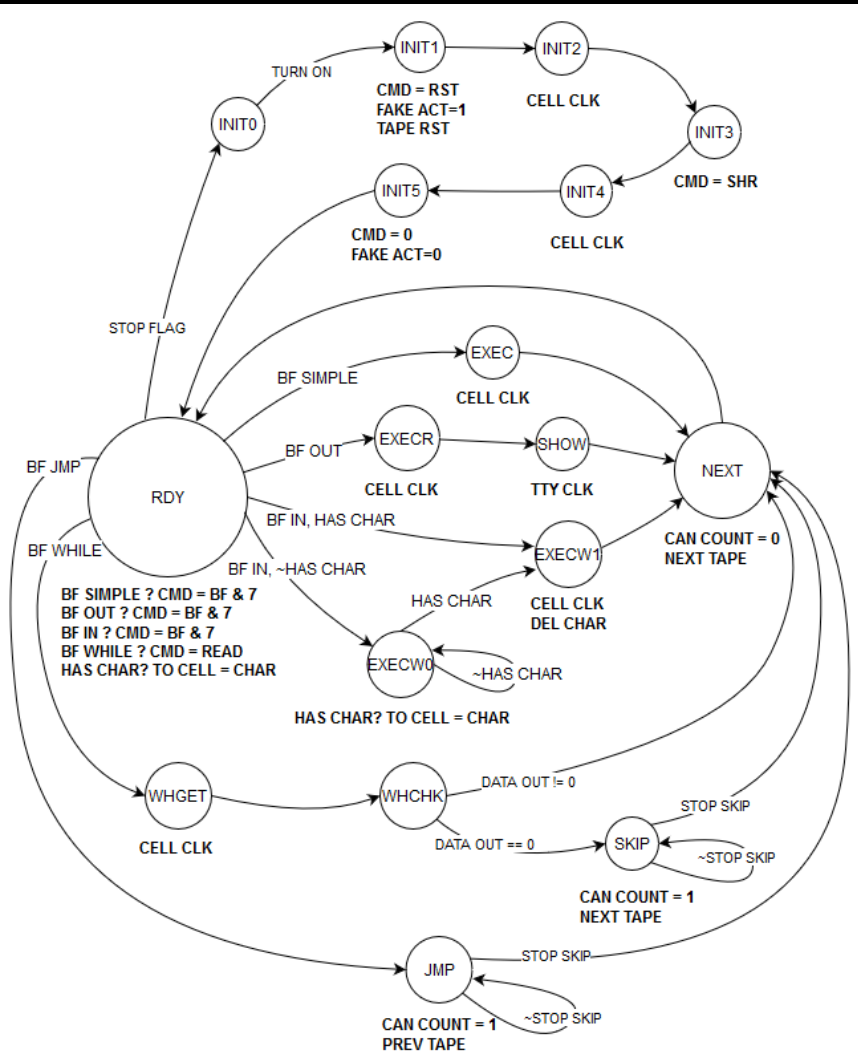


Команда END

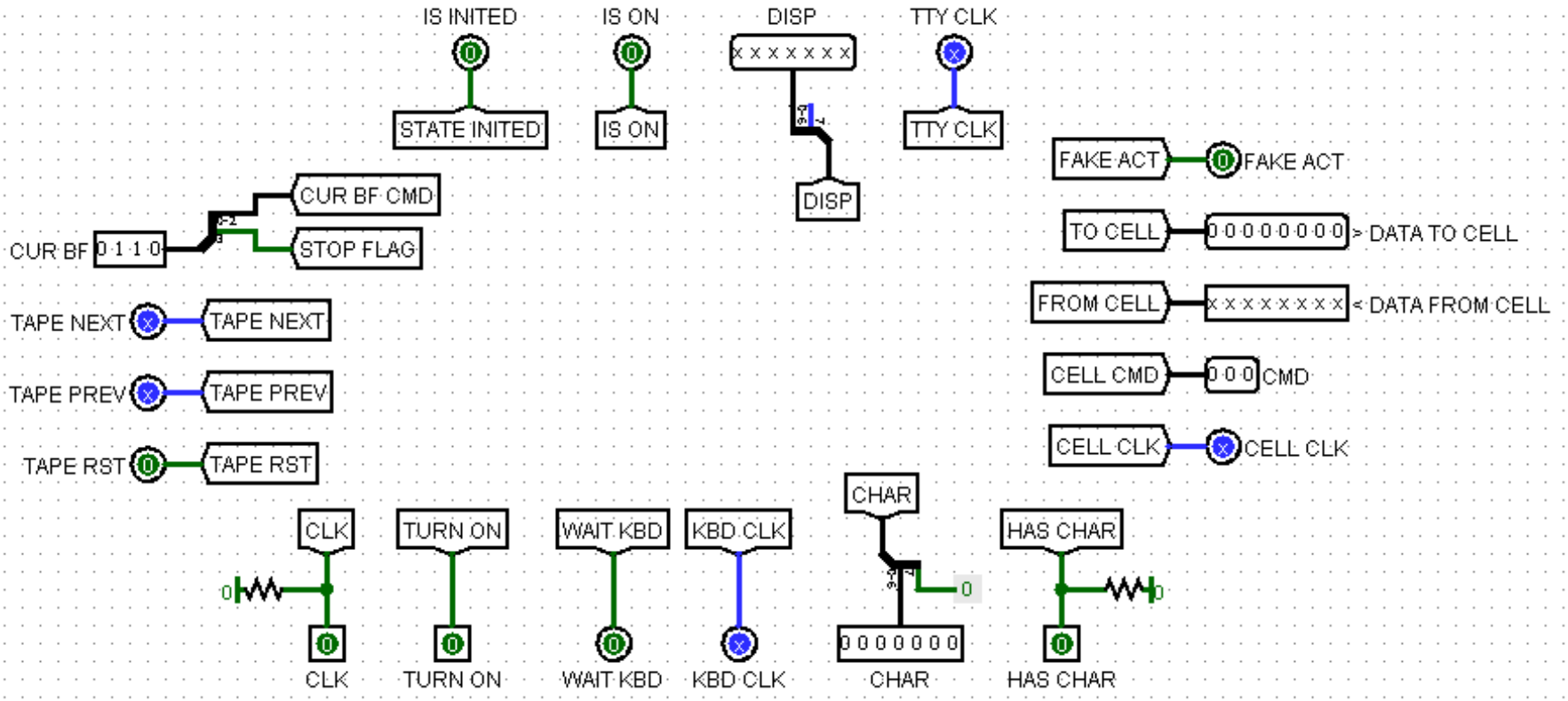
- Переводит автомат в исходное положение

Итог

- 17 состояний
- 27 переходов
- 22 бита входных сигналов
- 20 бит выходных сигналов



Выходы микросхемы процессора



Входные сигналы автомата

- TURN ON – сигнал включения
- HAS CHAR – сигнал наличия символа в буфере клавиатуры
- CHAR (7 бит) – текущий символ клавиатурного буфера
- DATA OUT (8 бит) – значение, прочитанное из текущей активной ячейки
- STOP SKIP – сигнал счетчика скобочной последовательности о том, что последовательность команд WHILE-JUMP сбалансирована
- BF CMD (3 бита) – закодированная команда Brainfuck
- STOP FLAG – лента указывает на команду END

Выходные сигналы автомата

- SET ACT – включает буферный D–триггер FAKE ACT позволяющий сделать активной головную ячейку при инициализации
- CLEAR ACT – выключает буферный D–триггер FAKE ACT
- TAPE RST – сигнал для перемотки ленты команд на начало
- CMD (3 бита) – команда для ячеек памяти, которая поступает на вход буферного регистра
- CELL CLK – посылка тактового сигнала в ячейки по шине
- TTY CLK – отправка тактового импульса в терминал
- DEL CHAR – сигнал удалить текущий символ из клавиатурного буфера
- NEXT TAPE – сигнал ленте перейти на следующую команду
- PREV TAPE – сигнал ленте перейти на предыдущую команду
- TO CELL (8 бит) – данные для ячейки памяти, которые поступают на вход соответствующего буферного регистра
- CAN COUNT – сигнал, разрешающий подсчет баланса скобочной последовательности WHILE-JUMP

Формулы выходных сигналов (1/2)

- SET ACT = IS INIT1
- CLEAR ACT = IS INIT5
- TAPE RST = IS INIT1
- NEXT TAPE = IS NEXT $\vee$ IS SKIP
- TTY CLK = IS SHOW
- DEL CHAR = IS EXECW1
- PREV TAPE = IS JMP
- CAN COUNT = IS JMP $\vee$ IS SKIP

Формулы выходных сигналов (2/2)

- "CELL CLK = IS INIT2 ∨ IS INIT4 ∨ IS EXEC ∨ IS EXECR ∨ IS EXECW1 ∨ IS WHGET
- TO CELL = CHAR если HAS CHAR ∧ (IS RDY ∨ IS EXECW0), то есть условие открывает ключ из CHAR в TO CELL
- $$\mathrm{CMD} = \begin{cases} \text{BF CMD}, \text{если IS RDY} \wedge \neg \text{IS LIKE JMP CMD} \\ \text{CMD\_RST}, \text{если IS INIT1} \\ \text{CMD\_SHR}, \text{если IS INIT3} \\ 0, \text{если IS INIT5} \\ \text{CMD\_READ}, \text{если IS RDY} \wedge (\text{BF CMD} == \text{BF\_WHILE}) \end{cases}$$

Функция возбуждения элементов памяти

- 17 состояний, невозможно построить формулу
- Декодер текущего состояния создает сигналы «IS состояние»
- Каждому состоянию соответствует битовая константа Logisim
- Константы с номерами состояний открываются ключами согласно таблице перехода

Таблица переходов (1/3)

Условие ключа	Константа нового состояния
IS INIT0 & TURN ON	ST_INIT1
IS INIT1	ST_INIT2
IS INIT2	ST_INIT3
IS INIT3	ST_INIT4
IS INIT4	ST_INIT5
IS INIT5	ST_RDY
IS RDY & IS SIMPLE CMD	ST_EXEC
IS RDY & (BF CMD == BF_OUT)	ST_EXECR

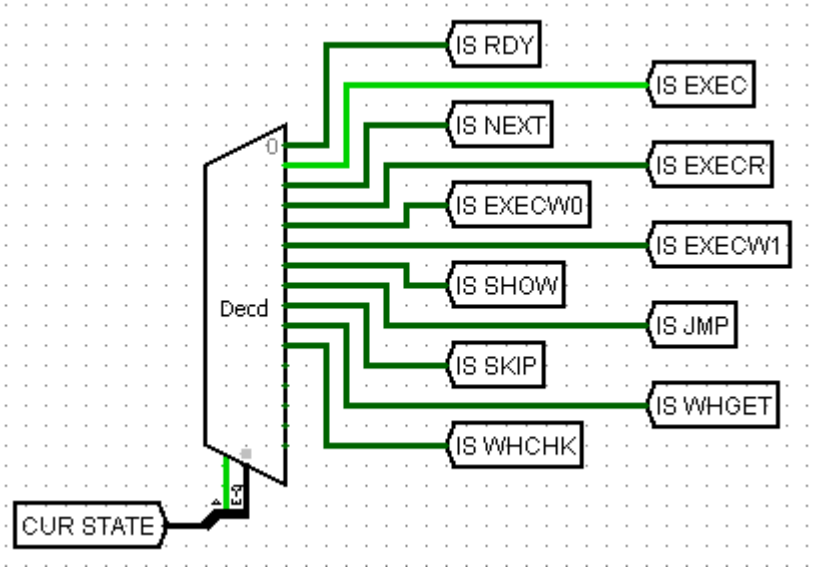
Таблица переходов (2/3)

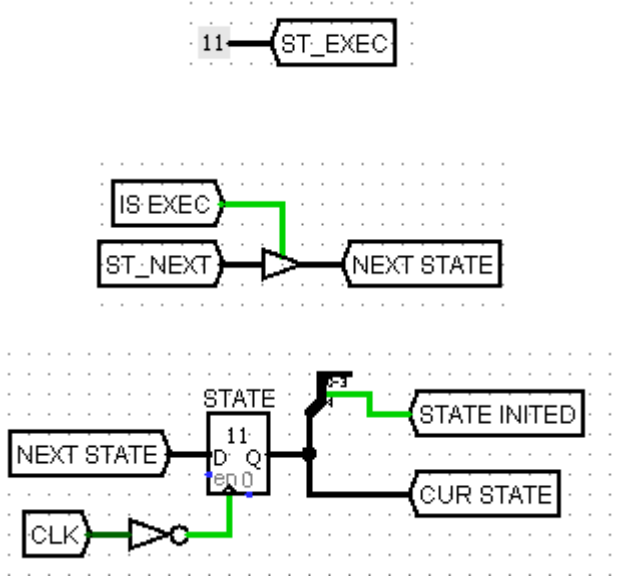
Условие ключа	Константа нового состояния
IS RDY & (BF_CMD == BF_IN) & HAS CHAR	ST_EXECW1
IS RDY & (BF_CMD == BF_IN) & ~HAS CHAR	ST_EXECW0
IS RDY & (BF_CMD == BF_WHILE)	ST_WHGET
IS RDY & (BF_CMD == BF_JMP)	ST_JMP
IS EXEC	ST_NEXT
IS EXECR	ST_SHOW
IS SHOW	ST_NEXT
IS NEXT	ST_RDY

Таблица переходов (3/3)

Условие ключа	Константа нового состояния
IS EXECW0 & HAS CHAR	ST_EXECW1
IS EXECW1	ST_NEXT
IS WHGET	ST_WHCHK
IS WHCHK & CELL ZERO	ST_SKIP
IS WHCHK & ~CELL ZERO	ST_NEXT
IS SKIP & STOP SKIP	ST_NEXT
IS JMP & STOP SKIP	ST_NEXT
IS RDY & STOP FLAG	ST_INIT0

Реализация схемы переходов (1/24)





Вопросы?