

Моделирование компьютерных систем с помощью SMV

Занятие №12

Час работы научит большему, чем день объяснений

Ж.-Ж. Руссо.

Александр Сергеевич Камкин

kamkin@ispras.ru

Каскадный сумматор (ripple-carry adder)

- Реализуйте *каскадный сумматор k -битных слов ($k = 4$)*
 - x и y – складываемые k -битные слова
 - z – результат сложения (k -битное слово)
 - C_{out} – выходной бит переноса
- Используйте *полные сумматоры (full adders)*:
 - C_{in} – входной бит переноса
 - x и y – складываемые биты
 - z – бит результата
 - C_{out} – выходной бит переноса

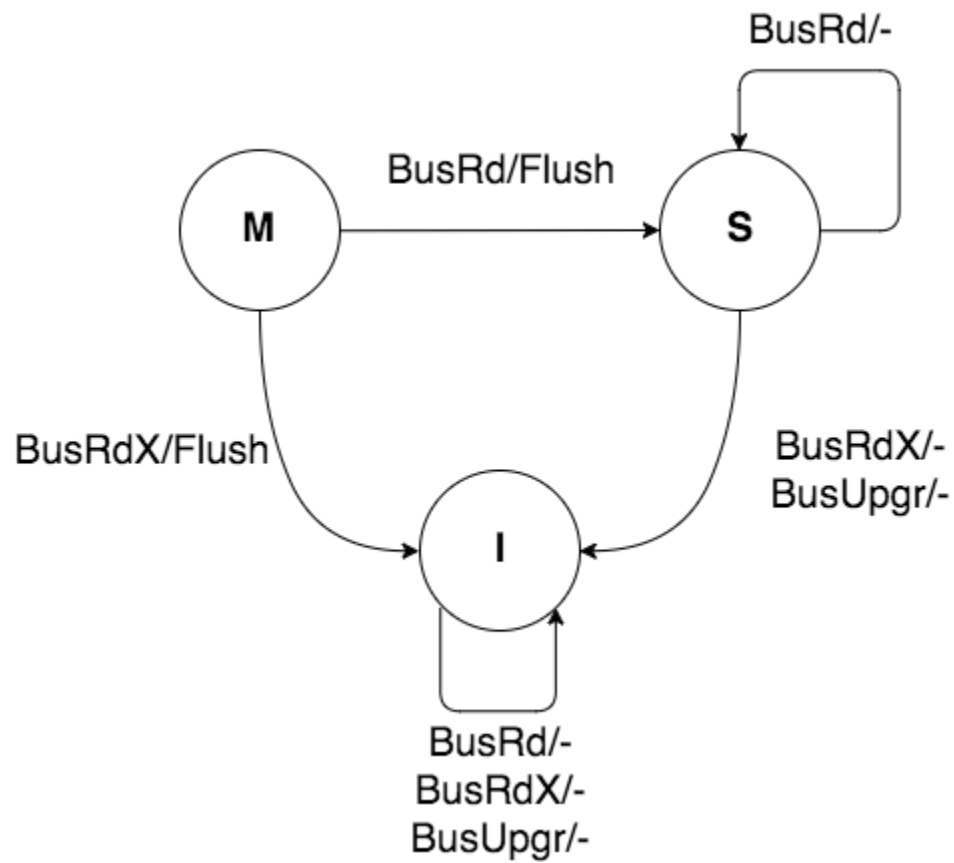
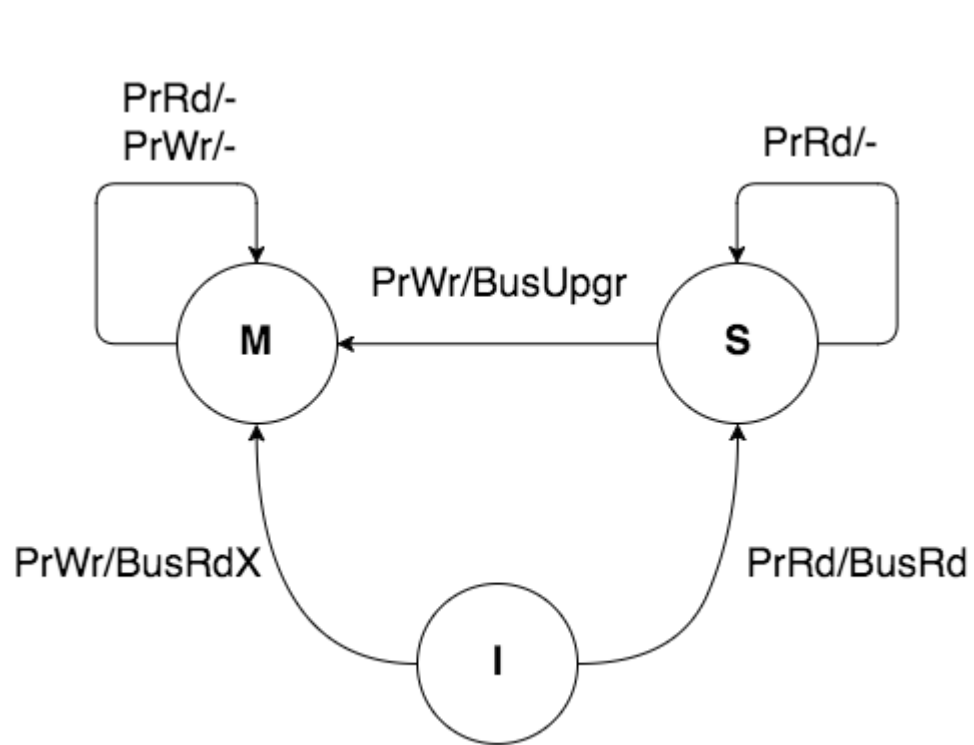
Очередь (FIFO) ограниченного размера

- Реализуйте *очередь из N ($N = 8$) k -битных слов ($k = 4$)*
- Возможность параллельного добавления и извлечения
- enq_{val} – вход: признак добавления (1 бит)
- enq_{dat} – вход: записываемое слово (k бит)
- deq_{val} – вход: признак извлечения (1 бит)
- deq_{dat} – выход: извлекаемое слово (k бит)
- enq_{rdy} – выход: признак возможности добавления (1 бит)
- deq_{rdy} – выход: признак возможности извлечения (1 бит)

Протокол обеспечения когерентности MSI (1)

- Опишите обобщенный протокол обеспечения когерентности MSI для N -ядерной системы ($N = 4$)
 - Состояния кэш-строки
 - M (*modified*) – модифицирована ядром
 - S (*shared*) – разделяется некоторыми ядрами
 - I (*invalid*) – отсутствует в кэш-памяти
 - Моделируемые операции
 - *read* – чтение данных из памяти
 - *write* – запись данных в память
 - *evict* – вытеснение кэш-строки

Протокол обеспечения когерентности MSI (2)



Практикум №2

- Реализуйте на языке C++ программу построения ROBDD по формуле булевой функции для заданного порядка переменных
 - Интерфейсы: см. `formula.h` и `bdd.h`
 - Срок: 14 мая