

PEC

# **Entrega Inicial Processador**

Andrea Querol, Pablo Vizcaino

April 2, 2019

## 1 Instruccions

- **Operacions lògiques i aritmètiques:** funcionen correctament.
- **Comparació amb i sense signe:** funcionen correctament.
- **Add immediat:** funcionen correctament.
- **Load:** funcionen correctament.
- **Store:** funcionen correctament.
- **Moure immediat:** funcionen correctament.
- **Salt condicional mode relatiu al PC:** funcionen correctament.
- **Extensió aritmètica:** funcionen correctament.
- **Ruptura de seqüència mode registre:** funcionen correctament.
- **Load Byte:** funcionen correctament.
- **Store Byte:** funcionen correctament.

## 2 Controlador de memòria

Tot i que per la lectura només necessitem un únic estat, per l'escriptura en necessitem mínim dos (la senyal SRAM\_WE\_N ha d'estar baixada una estona abans d'ocupar el bus amb la dada a escriure). El que hem fet nosaltres és tenir dos estats d'escriptura:

- Esc0: només baixa la senyal mencionada.
- Esc1: ocupa el bus amb l'escriptura, realitzant-la.

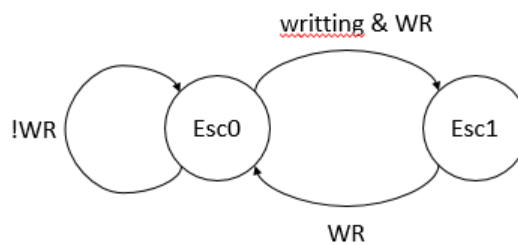


Figure 1: Graf d'estats d'escriptura del *SRAMController*.

Quan no hi ha permís d'escriptura ( $WR=0$ ) desactivem el canvi entre estats, de manera que ens quedem a Esc0 esperant. Per altra banda, preparem la senyal "writing" a 1, de manera que quan WR valgui 1, es produirà un canvi d'estat cap a Esc1 i es desactivarà aquesta senyal. Writing romandrà a 0 mentre WR no sigui igual a 0. Després d'un cicle en Esc1 sempre canviarà a Esc0, ja que WR seguirà valent 1 i és un canvi d'estat incondicional. Durant el que queda d'escriptura, l'estat es mantindrà a Esc0 perquè writing=0. Una vegada s'acabi l'escriptura, writing tornarà a valdre 1 ( $WR=0$ ) per a preparar-ne la següent.

Aquest sistema es podria simplificar afegint un tercer estat "Idle" mentre no hi haguessin escriptures o ja s'hagués realitzat la escriptura.