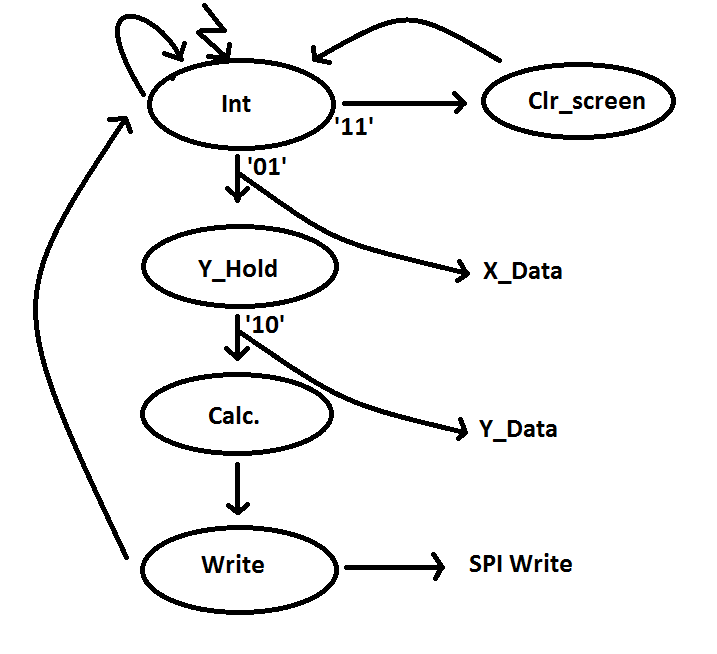
**GPU**

De GPU is ontworpen aan de hand van een FSM, deze bevat 5 states en staat hieronder weergegeven.



In de eerste state kan de GPU drie richtingen op gaan. Als de eerste twee bits van de 8-bits vector '11' zijn gaat de GPU naar Clr\_screen, hier zal de VGA zijn scherm leeg maken. De GPU gaat zijn X\_data inlezen en springt naar de volgende state (Y\_Hold) als de eerste twee bits van de 8-bits vector '01' is. Voor iedere andere waarde op de eerste twee bits van de 8-bits vector blijft de GPU in de int state.  
Vanuit de Clr\_screen state kan de GPU alleen nog maar terug naar de int state en begint de FSM opnieuw.  
In de Y\_Hold state wacht de GPU op Y\_Data, de GPU blijft in deze state totdat de eerste twee bits van de 8-bits vector '10' aangeven (Duurt 20 klokslagen). Op dat moment leest de GPU de Y\_Data in en stapt over naar de Calc. state.   
In deze state wordt het adres berekend voor de VGA, dit gebeurd met de formule 32\*Y+X = adres. Als deze berekening voor het bepalen van zijn adres is voltooid springt de GPU automatisch naar de volgende state, de Write state. In de Write state schrijft de GPU data uit de buffers weg via SPI. Automatisch zal de GPU terug springen naar zijn eerste state, het proces begint dan opnieuw.