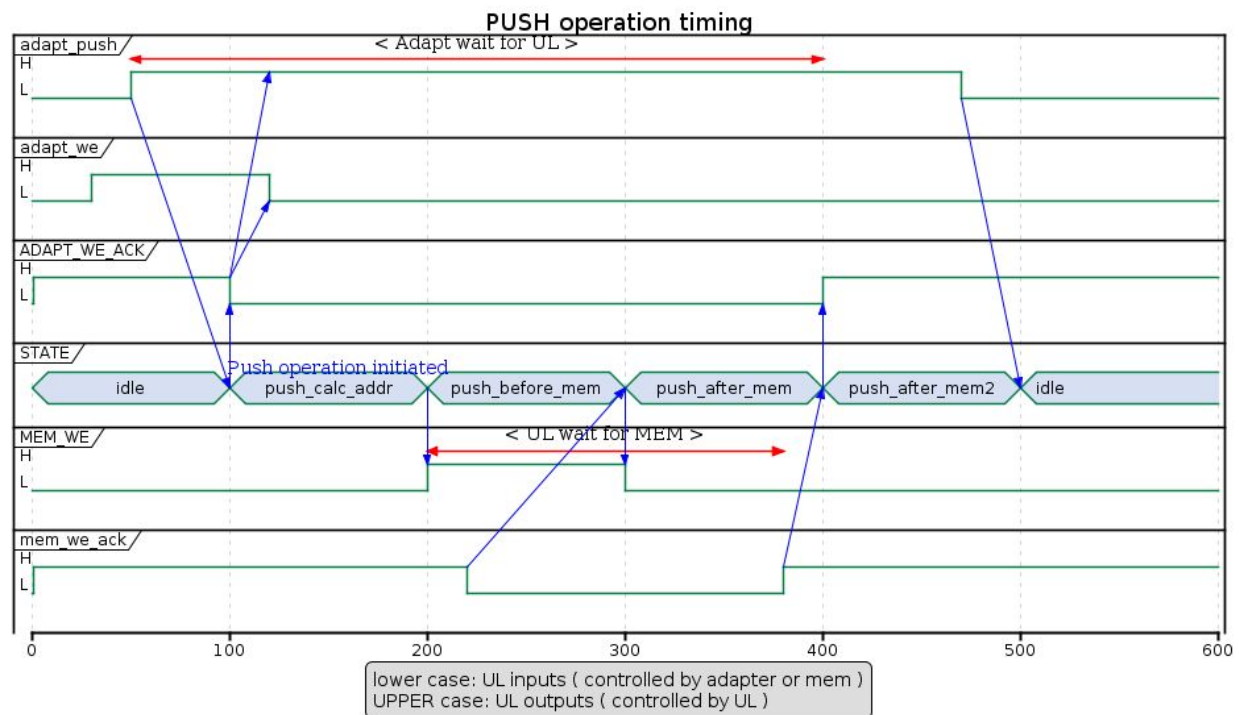


ACU Stack vezérelő

A modul interfészeinek időzítési modellje

(kivéve az ACU MMIO interfészt).

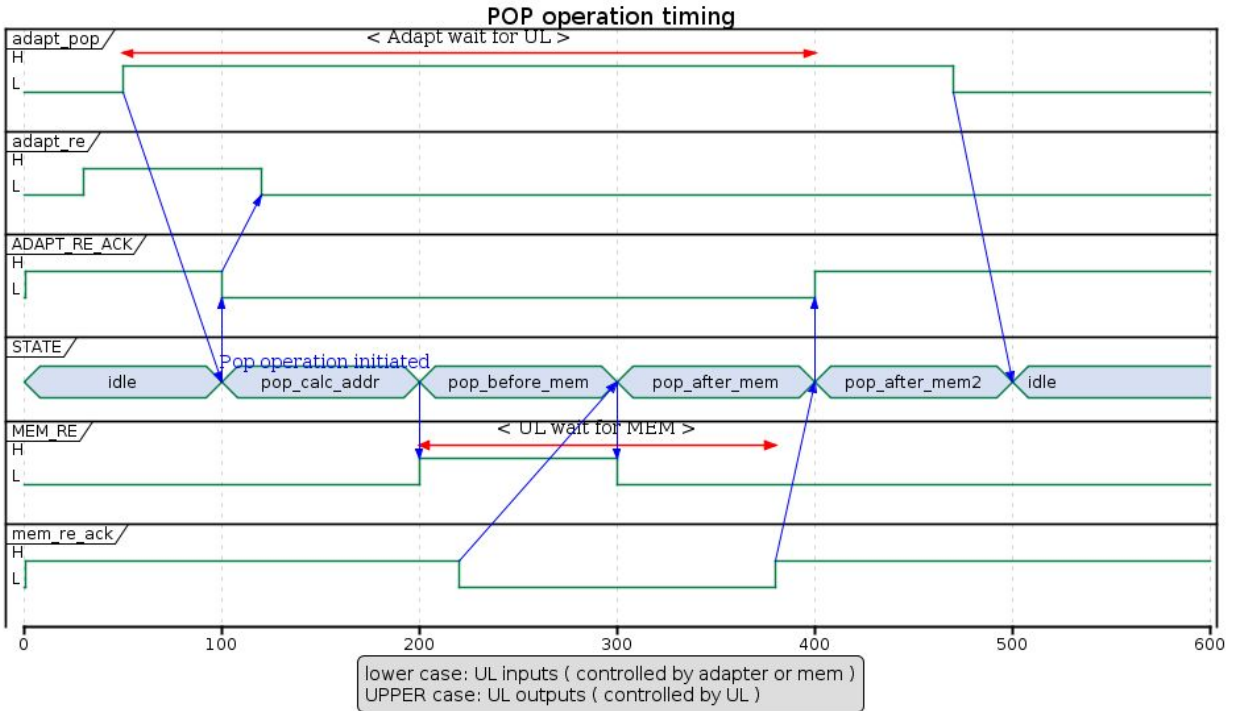
PUSH művelet esetén:



1. Ha **idle** állapotban a PUSH cím van kiválasztva és az írás engedélyezett (**adapt_push** = '1' and **adapt_we** = '1'), akkor a verem logika az **adapt_we_ack** jel alacsonyba vitelével jelzi az adapter felé, hogy az írási műveletet megkezdte. Kiszámítja a következő verem bejegyzéshez tartozó memória címet, az adapter felől kapott adatot továbbítja a memória felé és átvált a **push_calc_addr** állapotba.
2. **push_calc_addr** állapotban a memória felé **mem_we** jel magasba vitelével kéri egy új írási ciklus kezdetét, majd átvált **push_before_mem** állapotba. (Erre az állapotra azért van szükség, hogy a **mem_we** jel aktiválásakor már stabil adatok legyenek a memória cím és adat buszán.)
3. A **push_before_mem** állapotban addig várakozik, amíg a memória a **mem_we_ack** alacsony szintre húzásával nem jelzi, hogy az írási ciklust megkezdte.
4. A következő állapot a **push_after_mem**, ebben addig áll amíg a memória **mem_we_ack** újbóli magas szintjével nem jelzi, hogy az írási művelet befejeződött.

5. Az ezután következő **push_after_mem2** állapot addig áll fenn, amíg az MMIO adapter a PUSH cím aktivitását jelzi a verem logika felé.

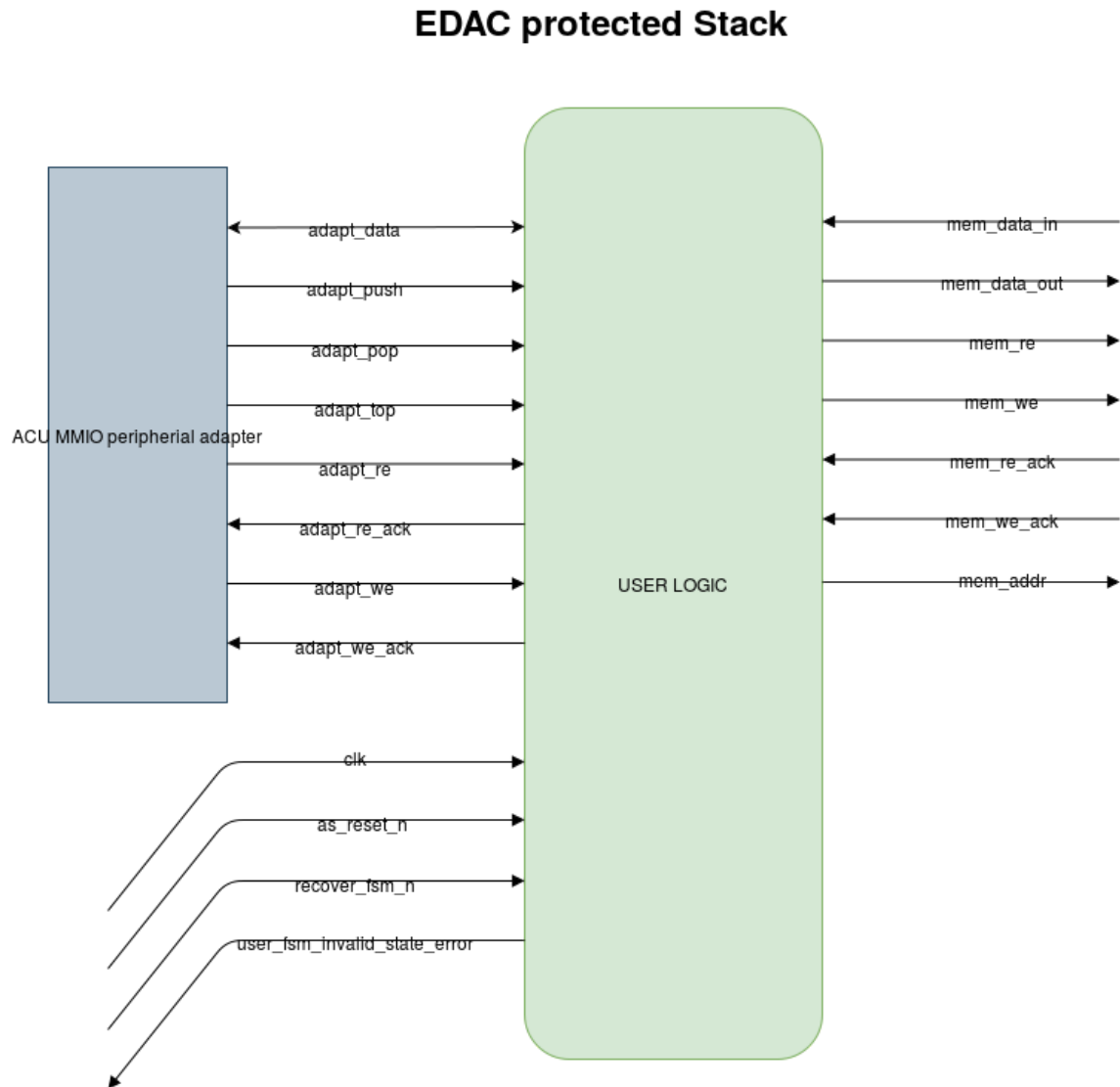
POP művelet esetén:



6. Ha **idle** állapotban a POP cím van kiválasztva és az olvasás engedélyezett (**adapt_pop** = '1' and **adapt_re** = '1'), akkor a verem logika az **adapt_re_ack** jel alacsonyba vitelével jelzi az adapter felé, hogy az olvasási műveletet megkezdte. A **stack_pointer**-ben levő címet kiadja a memória felé és átvált a **pop_calc_addr** állapotba.
7. **pop_calc_addr** állapotban a memória felé **mem_re** jel magasba vitelével kéri egy új olvasási ciklus kezdetét, majd átvált **pop_before_mem** állapotba. (Erre az állapotra azért van szükség, hogy a **mem_re** jel aktiválásakor már stabil legyenek a memória cím.)
8. A **pop_before_mem** állapotban addig várakozik, amíg a memória a **mem_re_ack** alacsony szintre húzásával nem jelzi, hogy az olvasási ciklust megkezdte.
9. A következő állapot a **pop_after_mem**, ebben addig áll amíg a memória **mem_re_ack** újbóli magas szintjével nem jelzi, hogy az olvasási művelet befejeződött.
10. Az ezután következő **pop_after_mem2** állapot addig áll fenn, amíg az MMIO adapter a POP cím aktivitását jelzi a verem logika felé.

A TOP művelet megegyezik a POP-al, csak a nem történik közben a **stack_pointer** változtatása.

A modul blokkdiagramja vagy RTL kapcsolási sémája.



Szintézis paraméterek

Név	Leírás
data_width	Azt adja meg, hogy hány bitből áll egy szó a stack-ben. Meg kell egyeznie az ACU adatbuszának méretével.
stack_size_log2	A stack maximális méretének a 2-es alapú logaritmusa

addr_push	Az a cím az MMIO buszon, amelyre történő írás a stack push műveletét indítja
addr_pop	Az a cím az MMIO buszon, amelyről történő olvasás a stack pop műveletét indítja
addr_top	Az a cím az MMIO buszon, amelyről történő olvasás a stack top műveletét indítja

A STACK-et megvalósító felhasználói logika portjai

Name	width	in/out	Side	Description
adapt_push	1	in	Adapter	Vezérlőjel, amely a PUSH műveletet jelzi
adapt_pop	1	in	Adapter	Vezérlőjel, amely a POP műveletet jelzi
adapt_top	1	in	Adapter	Vezérlőjel, amely a TOP műveletet jelzi
adapt_re	1	in	Adapter	Az adapter jelzi a User Logic számára egy olvasási műveletet kezdetét
adapt_we	1	in	Adapter	Az adapter jelzi a User Logic számára egy írási műveletet kezdetét
adapt_re_ack	1	out	Adapter	A User Logic jelzi az adapter felé, hogy olvasási művelet eredménye az adapt_data vonalakon rendelkezésre áll.
adapt_we_ack	1	out	Adapter	A User Logic jelzi az adapter felé, hogy az adapt_data vonalakon levő adatot feldolgozta.
adapt_data	data_width	in/out	Adapter	Kétirányú adatbusz az Adapter és a User Logic között
clk	1	in		Órajel

as_reset_n	1	in		Rendszer szintű reset vonal
recover_fsm_n	1	in		Alacsony-aktív vezérlő bemenet, amellyel az állapotgép hiba után alapállapotba állítható.
user_fsm_ivnvalid_state_error	1	out		Állapotgép érvénytelen állapotát jelző, magas-aktív állapotjel.
mem_data_out	data_width	out	Mem	A memória data bemeneti portja (connected to entity edac_protected_ram.data_in)
mem_data_in	data_width	in	Mem	A memória data kimeneti portja (connected to entity edac_protected_ram.data_out)
mem_re	1	out	Mem	A User Logic jelzi a memória számára egy olvasás műveletet kezdetét
mem_we	1	out	Mem	A User Logic jelzi a memória számára egy írási műveletet kezdetét
mem_re_ack	1	in	Mem	A memória jelzi a User Logic felé, hogy olvasási művelet eredménye az mem_data_in vonalakon rendelkezésre áll.
mem_we_ack	1	in	Mem	A memória jelzi a User Logic felé, hogy az mem_data_out vonalakon levő adatot feldolgozta.
mem_addr	stack_size_log2	out	Mem	A memória modulon belüli szó címe

A modulban megtalálható órajel-szinkronizáló elemek.

A áramkörben megvalósított metastabil szűrők mind double-flop szinkronizálóval vannak megoldva.

Az adapter_logic.vhd ban található metastabil szűrő:

- Az mmio perifériavezérlő interface-en keresztül jövő strobe jeleket szűri, illetve ez kiegészül a úgynevezett bypass filterrel, amivel kiválasztható, hogy használni akarjuk-e a szűrőt.

A stack_logic.vhd ban található metastabil szűrő:

- Reset szinkronizáló áramkör
- recover_fsm-nek mind a bemeneti jelét, mind a kimeneti ack-ot szűrjük

A modul becsült erőforrásigénye

A modul szintézisét a következő FPGA-ra végeztük:

Cyclone III EP3C16F484C6

Szintézis után a Quartus által generált "Resource Utilization by Entity report":

Compilation Hierarchy Node	LC Combinationals	LC Registers	Memory Bits	Pins
acu_stack	256 (26)	227 (4)	672	56
> edac_protected_ram:MEM	87 (11)	157 (8)	672	0
> > edac_memory_core:L_EDAC_PROTECTED_MEMORY	76 (27)	149 (75)	672	0
> > > altsyncram:memory_content_rtl_0	0 (0)	0 (0)	672	0
> > > > altsyncram_8ai1:auto_generated	0 (0)	0 (0)	672	0
> > > edac_decoder:L_DECODER	38 (38)	37 (37)	0	0
> > > edac_encoder:L_ENCODER	11 (11)	37 (37)	0	0
> edac_protected_stack:L_USER_LOGIC	143 (143)	66 (66)	0	0

(DSP és Virtual Pin komponens nem kerül felhasználásra a szintézis során.)

Azt látjuk, hogy a 672 bitnyi blokk memória került felhasználásra, ami 16 bites szóhossz és 32 szó stack méret mellett szavanként 5 bitnyi hibajavító adatot jelent.