

Dispositius Programables

Control - Novembre 2016

1. L'arquitectura AVR permet tenir una pila aprofitant la memòria de dades. Aquesta pila està construïda en la part alta (últimes adreces) de la memòria de dades. Hi ha instruccions màquina que aprofiten aquesta pila, per tant, la implementació de la pila està feta a partir del hardware de l'AVR.
 - a) Del joc d'instruccions, llista les instruccions màquina que fan servir aquesta pila hardware indicant com utilitzen la pila.
2. A part d'aquesta pila hardware, volem disposar d'una pila software que anomenarem *pila_tic*. Aquesta pila pot apilar bytes. La *pila_tic* la crearem reservant espai a la memòria de dades, amb una mida màxima en bytes que vindrà definida per una constant MAXP, de tal manera que si desitgem variar la mida d'aquesta pila, l'únic que s'hauria de fer és canviar aquest valor just abans de tornar a ensamblar el codi del nostre programa. Per controlar fins on està de plena aquesta pila definirem una variable a memòria de dades anomenada *i_pila* que consistirà en un byte indicant quants elements hi ha a la *pila_tic*.
 - a) Suposant que volem una *pila_tic* de 64 bytes, defineix en ensamblador *pila_tic* i *i_pila*. En quina secció de memòria s'ha de definir?
 - b) Quin pot arribar a ser el valor màxim que pot tenir MAXP amb l'estructura de dades que acabem de definir?
3. En l'apartat anterior hem aconseguit definir una estructura de dades que representa una pila. Però ara necessitarem definir les operacions (accions) que podem realitzar amb aquesta nova pila. Per tant, les operacions seran: a) **ini_p**: inicialitza al pila, es a dir, buida el contingut d'aquesta pila. b) **ppush**: inserta un element a la pila. c) **ppop**: treu un element de la pila. Aquestes tres operacions les implementarem com a tres subrutines. Aquestes subrutines han de ser transparents. Aquestes subrutines no han de comprovar possibles errors com des-apilar d'una pila buida o apila en una pila plena.
 - a) Defineix en pseudocodi (python) com serien les funcions **ini_p**, **ppush** i **ppop**.
 - b) Defineix la subrutina **ini_p** en ensamblador, tenint en compte que no té cap paràmetre ni retorna cap resultat.
 - c) Defineix la subrutina **ppush** en ensamblador, tenint en compte que el valor que es vol inserir a la pila es troba a r16. No retorna cap resultat.
 - d) Defineix la subrutina **ppop** en ensamblador, tenint en compte que no té cap paràmetre però retorna el valor des-apilat a r16
4. Una altra manera de definir l'estructura de dades d'aquesta pila consisteix en que la variable *i_pila* en comptes d'indicar el nombre d'elements que hi ha a la pila, indica l'adreça de memòria de dades on es troba l'últim element.
 - a) De quina mida és la variable *i_pila* en bytes? i per què?
 - b) Com podríem definir la variable *i_pila* a memòria de dades?
 - c) Defineix en pseudocodi (python) les funcions **ini_p**, **ppush** i **ppop** tenint en compte aquesta nova estructura de dades.
 - d) Redefineix la subrutina **ini_p** en ensamblador, tenint en compte aquesta nova estructura de dades.
 - e) Redefineix la subrutina **ppush** en ensamblador, tenint en compte aquesta nova estructura de dades.
 - f) Redefineix la subrutina **ppop** en ensamblador, tenint en compte aquesta nova estructura de dades.

5. Suposant que disposem d'aquesta nova pila i les seves operacions. Crea un programa en assemblador que fent servir la interrupció de recepció del port sèrie vagi apilant qualsevol caràcter enviat i el retorni pel port sèrie, però quan s'envii el caràcter 0x0a des-apili un element i el retorni pel port sèrie. Supposeu que disposeu de la subrutina TX. (1.5)
6. Defineix una subrutina *sumaN* que faci la suma amb precisió de N bytes de 2 operands. Els paràmetres d'aquesta *sumaN* són: el registre X com a l'adreça de memòria on es troba el primer operand, el registre Y com a l'adreça de memòria on es troba el segon operand i el registre r16 indicant els bytes de precisió. El resultat de la suma s'ha de deixar a la zona de memòria on es troba el segon operand. Els flags C, V, N i S també han de tenir els valors adequats un cop feta aquesta suma. Supposeu que els operands estan introduïts en format Little-endian. La subrutina ha de ser transparent. (2)