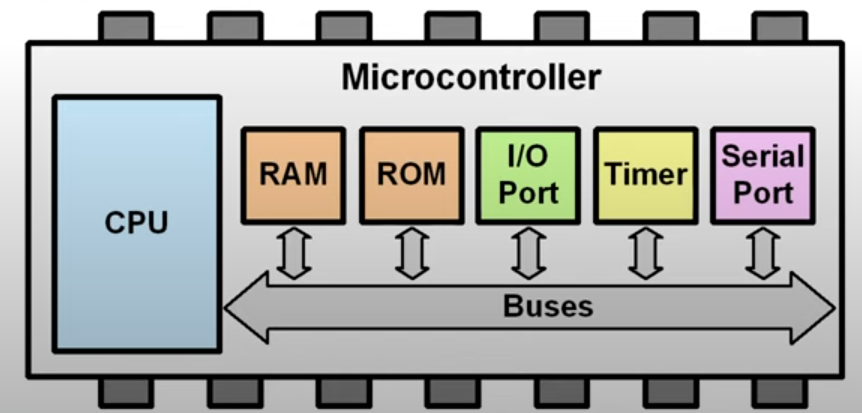
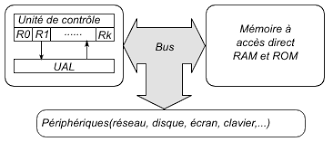
**CORRECTION**

**Q1. *Identifier*** *les fonctions du modèle de* ***von Neumann*** *sur le schéma du microcontrôleur.*



Les deux premières et les deux dernières lignes du fichier *Blink.hex* ont été réécrites ci-dessous.

*:10 0000* 00 **0C 94 5C 00 0C 94 6E 00 0C 94 6E 00 0C 94 6E 00** *CA*

*:10 0010* 00 **0C 94 6E 00 0C 94 6E 00 0C 94 6E 00 0C 94 6E 00** *A8*

…………

*:**0A 0420 00* ***1F 91 08 95 08 95 F8 94 FF CF*** *8E*

*:00000001FF*

*Consultez la ressource* [*https://urlr.me/nTsRw* pour répondre à Q2](https://urlr.me/nTsRw%20pour%20répondre%20à%20Q2) *à Q4.*

*Chaque ligne du fichier Blink.hex commence par :10.*

**Q2.** *À quoi correspond cette valeur ? À quoi correspondent les deux octets qui suivent (Exemple : 0000, 0010, etc) ?*

***BB = 1016 = 1610****: nombre d’octets du code machine dans la ligne du fichier .hex*

*AAAA correspond à l’adresse absolue (ou relative) du début de la ligne.*

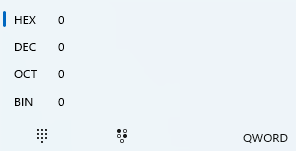
**Q3**. Combien d’octets contient le programme Blink ? Quelle est la taille maximale (en octets) qu’un ATMEMEGA328P peut exécuter ?

***AAAA = 42016*** *=* ***105610***

*1F16 est le 1057e code machine dans le fichier (première adresse = 0). Il en reste 9 avant sa fin. Le programme Blink contient donc 1066 octets.* L’ATMEGA328P dispose d’une mémoire flash de 32ko (16k x 16bits) soit 32768 octets.



*La ressource à consulter (* [*https://urlr.me/nTsRw*](https://urlr.me/nTsRw) *) donne les informations suivantes : « L'octet de checksum est le*[*complément à deux*](https://fr.wikipedia.org/wiki/Compl%C3%A9ment_%C3%A0_deux) *de la somme des valeurs binaires des octets de tous les autres champs. (Les calculs sont faits sur 8 bits, en ignorant les retenues.)*



**Q4**. Utilisez la calculatrice Windows en mode programmeur pour calculer le cheksum de la suite de valeurs hexadécimales : *0A 04 20 00* ***1F 91 08 95 08 95 F8 94 FF CF****. Quel réglage avez-vous appliqué ?*

*A + 4 + 20 + 1F + 91 + 8 + 95 + 8 + 95 + F8 + 94 + FF + CF = ~~5~~72 On conserve 72*

*Cheksum = complément à 2 de 7216 or 7216 = 011100102*

*Cheksum = 100011012 + 1 = 100011102 =* ***8E16***

**Q5**. Identifiez l’instruction assembleur correspondant à 21 E4 en mémoire.

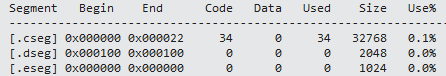
*E42116 = 1110 0100 0010 00012*

*- KKKK = 0100 (quartet de poids fort de K) donc* ***K*** *= 010000012 =* ***0x41***

*- dddd = 00102 = 210 de* ***d = 2*** *on peut déduire que le registre*

*- KKKK = 0001(quartet de poids faible de K) concerné est le deuxième après R16 soit* ***R18***

*L’instruction assembleur est* ***LDI R18,0x41***



**Q6**. Comparez l’empreinte mémoire du programme Blink avec celle de Blinkcp.

*Le programme Blink.cpp occupe 1066o alors que Blink.asm en occupe 34.*

*On perd la simplicité d’écriture offerte par les langages de haut niveau, mais le code écrit en assembleur est plus optimisé.*

**Q7.** **T(s) = 0,5.10-6.a.b** **r27r26 ← a** **r25r24 ← b**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **F(Hz)** | **T(s)** | **a.b** | **r27r26** | **r25r24** | **Remarque** |
| *0,5* | *2* | *4 .106* | *1000* | *4000* | *La led s’éclaire 1 fois toute les 2s pendant 1s.* |
| *0x03E8* | *0x0FA0* |
| 1 | 1 | *2 .106* | *500* | *4000* | *La led s’éclaire 1 fois par seconde pendant 0,5s.* |
| *0x01F4* | *0x0FA0* |
| 2 | 0,5 | *1 .106* | *500* | *2000* | *La led s’éclaire 2 fois par seconde pendant 0,25s.* |
| *0x01F4* | *0x07D0* |
| 4 | 0,25 | 0,5*.106* | *500* | *1000* | *La led s’éclaire 4 fois par seconde pendant 0,125s.* |
| *0x01F4* | *0x03E8* |