PIC VS. AVR Microcontroller

PIC

- Programmable Intelligent Computer"
- or
- "Peripheral Interface Controller"

AVR

AVR derives its name from its developers and stands for

- **Alf-Egil Bogen**
- **Vegard Wollan**
- **RISC microcontroller**,

- هذا السؤال دائما ما يتبادر لكل من يعمل أو بدأ يدخل مجال الأنظمة المدمجة، دائما سنجد هذا الصراع القائم بين فريق متحمس للـ AVR وآخر متحمس للـ PIC

في البداية لنعترف بشيء هام، في مجال النظم المدمجة لا يوجد ما يسمى "ما هو أفضل مُتحكِم دقيق" بصورة مُطلَقة ولكن هناك "من الأنسب" للاستخدام في تطبيق معين

- في بعض الأحيان نحتاج أن نصمم نظام تحكم بسعر رخيص جداً ولا نحتاج لقدرات خارقة أو متحكمات متطورة لتشغيله لذا نبحث عن المتحكم" الأرخص "والذي يكفي فقط لهذه المهمة
- لذا لا تستغرب أن المتحكمات STM8 (8-bit) تعتبر من أكثر المتحكمات مبيعاً في العالم لأنها أرخص من كل من PICوالـ AVR الـ 8 بت وتتفوق عليهم في تقديم قدرات مناسبة بسعر منخفض.

فمن الأفضل PIC أم AVR ؟؟؟

- للإجابة بعض المقارنات التقنية بين متحكمات كل من
- PIC – 8 bit و AVR – 8 bit

أولاً: مقارنة السرعة

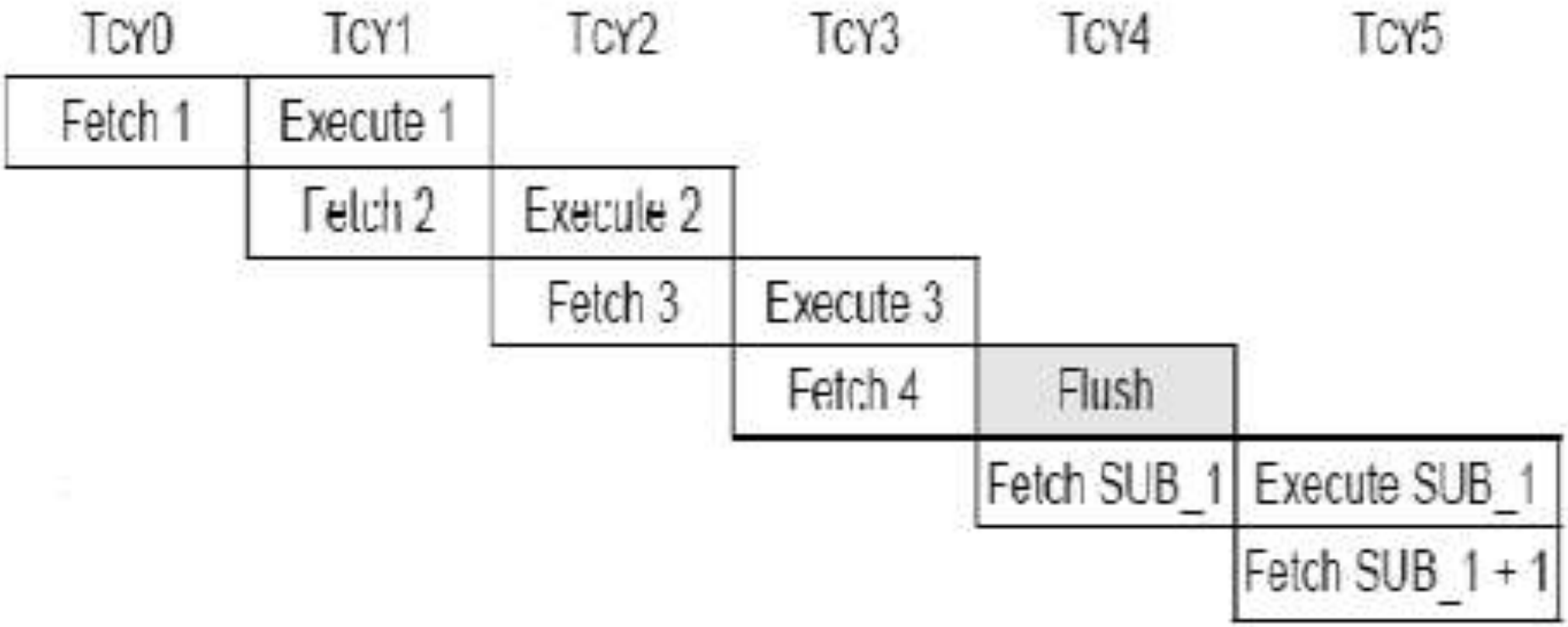
- هنا سنجد أن متحكمات الـ AVR – 8 bit تتفوق بفارق كبير جدا ويعتبر أدائها أسرع بنحو 4 أضعاف من مثيلتها في الـ PIC – 8 bit
- لأن متحكمات الـ AVR تستطيع أن تنفذ:

عدد أوامر في الثانية الواحدة = التردد الذي تعمل به

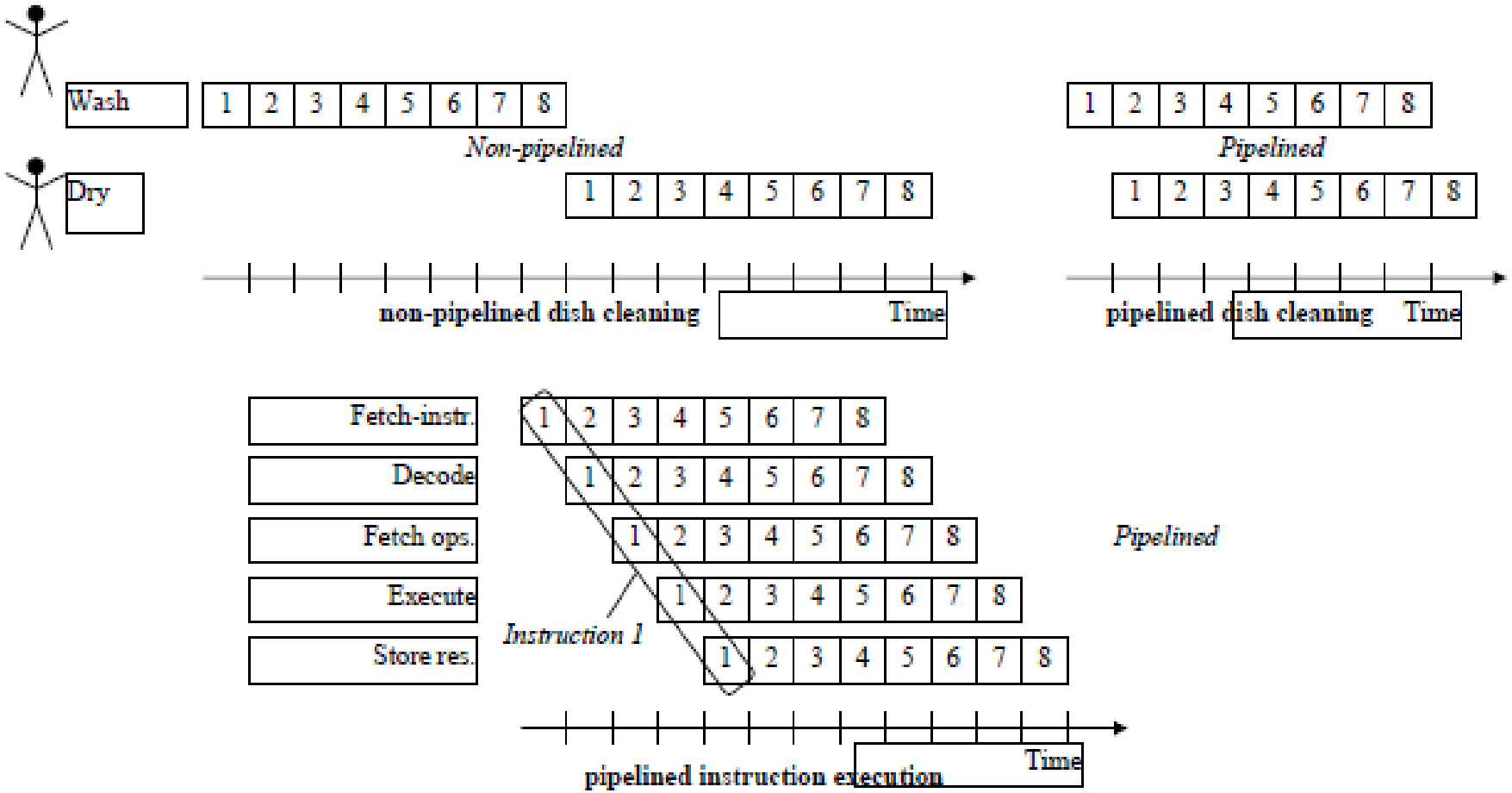
- أما الـ PIC فيمكنه تنفيذ ربع هذا العدد

- مثلا لو معنا متحكم PICو AVR
وكلاهما يعمل بتردد 16 ميجاهرتز (16 مليون هرتز)
- سنجد أن الـ AVR يمكنه تنفيذ 16 مليون أمر برمجي في الثانية الواحدة Instruction per second
- بينما الـ PICبنفس السرعة يستطيع أن ينفذ فقط 4مليون أمر في الثانية الواحدة.
- يرجع هذا الأمر إلى تقنية الـ Pipeline التي تتميز بها جمع متحكمات الـ AVR ولا تتواجد إلا في بعض فئات الـ PIC المتطورة نسبيا

Pipeline Technology?



Pipelining: Increasing Instruction *Throughput*



- إضافة إلى أن معظم شرائح الـ AVR تحتوي أيضاً على بعض الأدوات التي تسرع من تنفيذ الأوامر مثل الـ Hardware Multiplier
- وهي وحدة معالجة لعمليات الضرب الحسابية فيمكنها تنفيذ عملية الضرب في 2 نبضة فقط
- بينما في متحكمات الـ PIC المماثلة لا تحتوي على هذا الأمر وقد تستغرق نفس عملية الضرب نحو 40 ضعف الوقت المطلوب على الـ AVR

ثانيا :التصميم الداخلي ومعالجة البيانات

- عندما نكتب برنامج بلغة التجميع Assembly نجد فارقا ضخماً بين كليهما
- حيث يتمتع الـ AVR بوجود 32 مسجل عام Register يمكن استخدامهم في معالجة وتخزين البيانات المؤقتة بسرعة وكفاءة
- بينما يجبرك الـ PICعلى استخدام مسجل واحد فقط هو المكدس في معظم الأوامر
- وهذا يعني أن البرامج المكتوبة على الـ AVRأكثر كفاءة وأسرع بكثير من البرامج المكتوبة على الـ PIC

- مثال على ذلك، البرنامج التالي مكتوب بلغة السي ومصمم لكي يبحث عن أكبر قيمة داخل مصفوفة من الأرقام
- وتم تشغيل نفس الكود على مجموعة من المتحكمات الدقيقة
- ATmega16, PIC18F, MSP

```
int max(int *array)
{
 char a;

 int maximum=-32768;

 for (a=0;a<16;a++)

        if (array[a]>maximum)

                maximum=array[a];
return (maximum);
}
```

Device	Max Speed [MHz]	Code Size [Bytes]	Cycles	Execution Time [uS]
ATmega16	16	32	227	14.2
MSP430	8	34	246	30.8
PIC18F452	40	92	716	17.9

ثالثاً :استهلاك الطاقة

- هنا يتفوق الـ PIC على الـ AVR بفارق واضح
- فتتميز متحكمات الـ PIC باستهلاك منخفض للطاقة مقارنة مع الـ AVR
- على الرغم من محاولات شركة ATMEL من تحسين بعض إصدارات الـ AVR بتقنيات استهلاك منخفض بقيت الـ PIC أفضل في هذا الجانب

رابعاً البرمجة

- اعتمدت شركة Atmel على مترجمات Compilers مفتوحة المصدر وتدعم ANSI-C مثل الـ AVR-GCC مما تسبب في جعلها الخيار المفضل
- وهذا السبب الذي جعل مصممو لوحات الآردينو يختارون شرائح AVR بدلا من PIC
- أما شركة Microchip فقد اتخذت مساراً مختلفاً فنجد برنامج MPLAB يخالف الـ ANSI-C خاصة عند كتابة برامج لعائلات PIC16F مما يجعل تعديل الأكواد المكتوبة بها لاستخدامها مرة أخرى أو نقلها لمتحكمات أخرى عملية صعبة

- فهنا بتفوق الـ AVR

نستنتج من ذلك أنه في حالة الرغبة بتطوير نظام سريع الاستجابة أو يقوم بعمليات حسابية معقدة وبسعر مناسب فإن الـ AVR هو الخيار الأمثل لأغلب الأنظمة المدمجة المعتمدة على المُتحكِمات 8 بت الرخيصة

خامساً: السعر مقارنة بالمميزات المدمجة