

Programmed I/O

ثلاثة تقنيات ممكنة لـ I/O operations .

مع الـ Programmed I/O يتم تبادل البيانات بين المعالج و I/O Module . يقوم المعالج بتنفيذ برنامج يمنحه التحكم المباشر في I/O operation منها حالة الجهاز وارسال امر قراءة او الكتابة للبيانات و نقل البيانات. عندما يصدر المعالج امرا الى I/O Module يجب عليه الانتظار حتى تعتمل عملية الـ I/O . اذا كان المعالج اسرع من الـ I/O Module فهذا يعتبر وقت ضائع للمعالج. باستخدام interrupt-driven I/O يصدر المعالج امرا I/O ويستمر في تنفيذ في تنفيذ تعليمات اخرى وتتم مقاطعته بواسطة I/O Module عندما تنتهي من عملها الاخير.

مع الاثنين الـ programmed and interrupt I/O يكون المعالج مسؤولا عن استخراج البيانات من الذاكرة الرئيسية للـ output ويخزن البيانات في الذاكرة الرئيسية للـ input .

ويعرف البديل لهذا هو (DMA) direct memory access .

هو تقوم الـ I/O Module و الـ main memory بتبادل البيانات مباشرة دون تدخل المعالج.

	No Interrupts	Use of Interrupts
I/O-to-memory transfer through processor	Programmed I/O	Interrupt-driven I/O
Direct I/O-to-memory transfer		Direct memory access (DMA)

هذا الجدول الى العلاقة بين هاذي التقنيات.

Overview of Programmed I/O

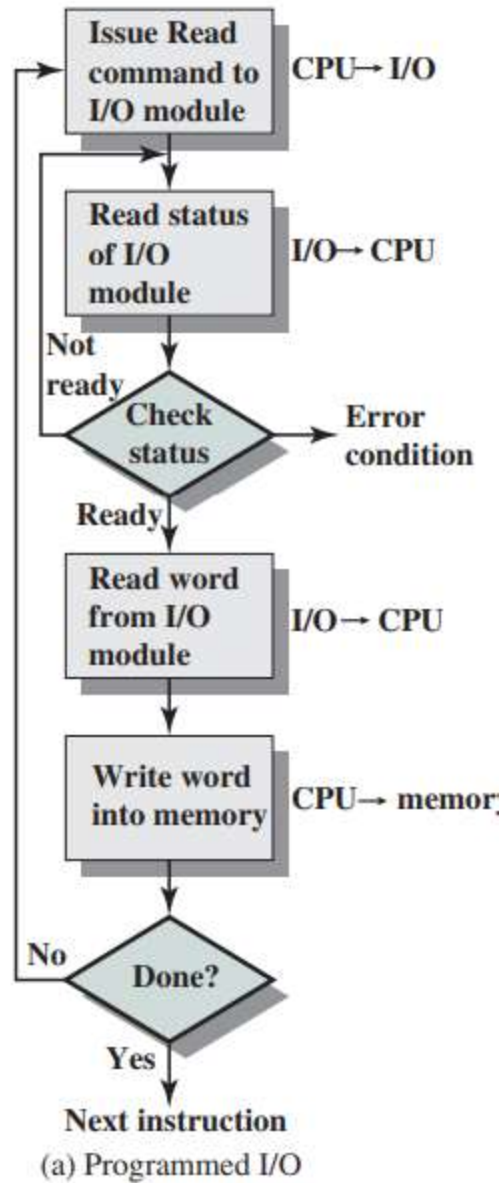
عندما يقوم المعالج بتنفيذ البرنامج ويواجه تعليمات تتعلق بالـ I/O فانه ينفذ تلك المعلومات عن طريق اصدار امر الى I/O Module المناسبة. مع الـ programmed I/O ستقوم I/O Module بتنفيذ الاجراء المطلوب ثم تعيين البتات المناسبة في I/O status register . لا تتخذ الـ I/O Module اي اجراء اخر لتنبيه المعالج وانه لا يقاطع المعالج ايضا.

وبتالي تقع على عاتق المعالج المسؤولية التحقق بشكل دوري من حالة الـ I/O Module حتى يجد ان العملية قد اكتملت. لشرح تقنية الـ programmed I/O ننظر اليها اولا من وجهة نظر I/O Commands التي يصدرها المعالج الى I/O Module ثم من وجهة نظر I/O instructions التي ينفذها المعالج.

I/O Commands

لتنفيذ تعليمات متعلقة بـ I/O يصدر المعالج عنوانا يحدد I/O Module المعينة والجهاز الخارجي و I/O Command . هناك اربعة انواع من الـ I/O Command التي قد تتلقاها I/O Module عندما تتم معالجتها بواسطة المعالج :

- **Control** : التحكم يستخدم لتنشيط الجهاز الطرفي واخباره ما يجب فعله. على سبيل المثال قد يطلب من وحدة magnetic-tape اعادة لف سجل واحد او تحريكه للامام. وتم تصميم هاذي الاوامر لاجهزة طرفية معينة.
- **Test** : الاختبار يستخدم لاختبار حالات الحالة (status) المختلفة المرتبطة بـ I/O Module و الاجهزة الطرفية الخاص بها. سيرغب المعالج ايضا في معرفة ما اذا كانت الـ I/O Operation الاخيرة قد اكتملت وما اذا حدثت ايا اخطاء.
- **Read** : القراءة يؤدي الى حصول الـ I/O Module على عنصر بيانات من الجهاز الطرفي ووضعة في مخزن المؤتت الداخلي (internal buffer). يمكن بعد ذلك المعالج الحصول على عنصر عن طريق مطالبة الـ I/O Module بوضعة data bus .
- **Write** : تجعل I/O Module تاخذ عنصرا من البيانات من data bus ثم تقوم بعد ذلك بارسال عنصر البيانات هذا الى الجهاز الطرفي.



الصورة هاذي توضح مثال على استخدام الـ I/O Programmed لقراءة الكتلة من جهاز الطرفي الى الذاكرة.

بالنسبة لكل word تتم قراءتها يجب ان يظل المعالج في دورة التحقق من الحالة

(status-checking cycle)

حتى يحدد ان الكلمة متاحة في I/O Module data register .

وهذا العيب واضح تستغرق وقتا طويلا والمعالج غير فاضي لا مؤاخذة المعالج : انا شغال عند اهلك ؟

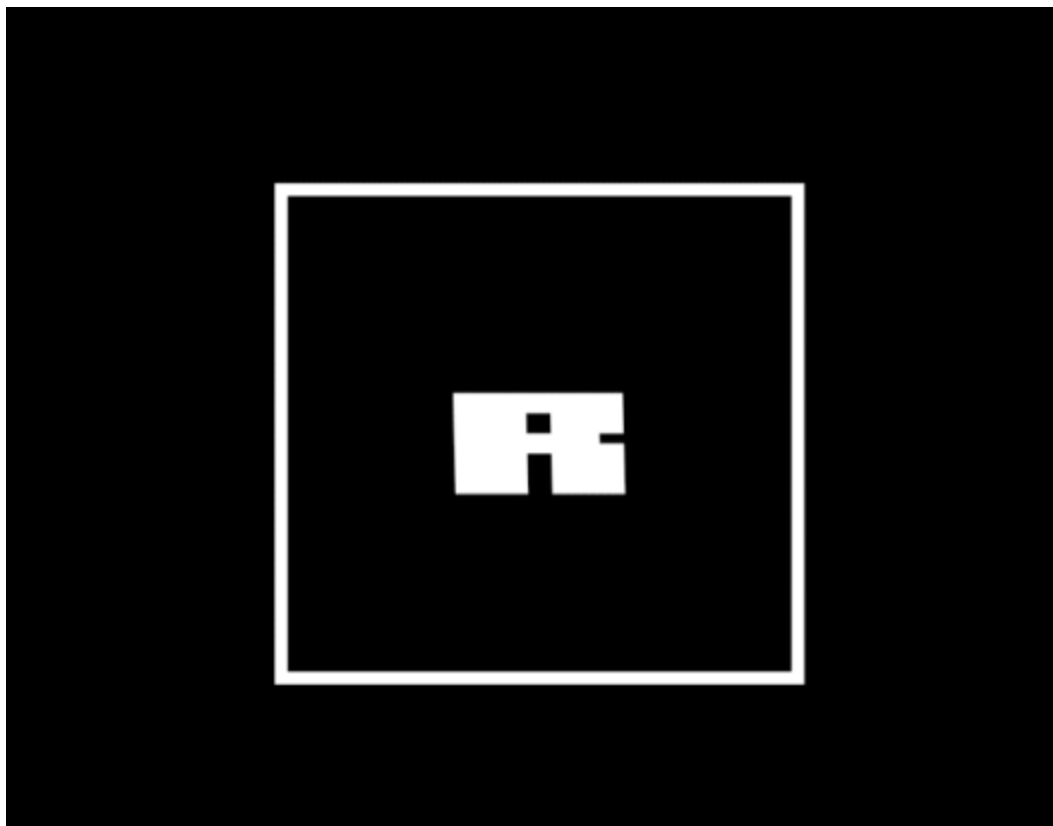
I/O Instructions

مع programmed I/O هناك توافق وثيق بين التعليمات المتعلقة بـ I/O التي يجلبها من الذاكرة واوامر I/O التي يصدرها المعالج الى I/O Module لتنفيذ التعليمات. وهذا يعني انه يتم تعيين التعليمات بسهولة في I/O . ويعتمد شكل التعليمات يعتمد على طريقة معالجة الاجهزة الخارجية.

عادة ما يكون هناك العديد من I/O Device المتصلة بالنظام من خلال I/O Module . يتم منح كل device عنوان فريدا. عندما يصدر المعالج I/O Command يحتوي الامر على عنوان الجهاز المطلوب. وبالتالي يجب على كل I/O Module ان تفسر سطور العنوان لتحديد ما اذا الامر خاصا بها ام لا. عندما يتشارك المعالج والذاكرة الرئيسية و I/O تتشارك في common bus يكون هناك وضعان ممكنان للعنونة : memory mapped and isolated .

يتعامل المعالج مع status and data registers الخاصة بـ I/O Module كمواقع للذاكرة ويستخدم نفس التعليمات machine للـ access الى كل من memory & I/O Device .

AhmadAlFareed



Twitter : https://twitter.com/dr_retkit

YouTube : <https://www.youtube.com/@retkit1823>