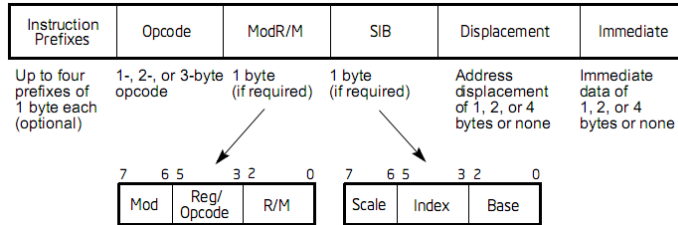


١ مقدمة عن أنظمة التشغيل

جهاز الحاسب هو مجموعة من الشرائح الإلكترونية والعتاديات والمتحكمات المرتبطة مع بعضها لتوفير منصة تشغيلية للبرامج والتي بدورها لن يعمل هذا الجهاز. ويمكن تقسيم البرامج بحسب طبيعة عملها ووظيفتها الى قسمين هما **برامج المستخدم** والتي صممت خصيصاً لحل مشاكل المستخدم و **برامج النظام** والتي تتحكم في عتاد وموارد الحاسب ، ويعتبر نظام التشغيل مثالا لبرامج النظام حيث يدير عتاد وموارد الحاسب بالإضافة الى ميزة مهمة وهي توفر بيئة تشغيل وهمية (Virtual Machine) لبرامج المستخدم. ويوضح التعريف السابق عدداً من المفاهيم التي يجب الوقوف عليها وتوضيحها بشكل مفصل. فجهاز الحاسب هو منصة تشغيلية حقيقية للأوامر ويأتي ذلك بسبب وجود متحكم خاص لمعالجة الأوامر وتنفيذها ، هذا المتحكم هو المعالج (Processor) حيث يعمل على تنفيذ الأوامر (من عمليات حسابية ومنطقية) وإرسال النتائج الى الأماكن المطلوبة. وتسمى مجموعة الأوامر التي ينفذها المعالج باسم **البرامج**، وبسبب تكلفة بناء المعالج فانه غالباً ما يتعرف على عدداً معيناً من الأوامر والتي تعرف بمجموعة الأوامر (Instruction Set). لذلك حتى يتم تنفيذ أوامر أي برنامج فانها يجب أن تُكتب وفقاً لمجموعة الأوامر التي يدعمها المعالج^١. والشكل ١.١ يوضح نموذجاً عاماً لتعليمات وأوامر المعالج التي تتكون منها البرامج. وجزءاً منها هي اختيارية وسنركز هنا على ال OPCODE والتي تمثل أوامر المعالج.

شكل ١.١: الشكل العام لأوامر المعالج x86



وتشكل أوامر المعالج لغة برمجية من خلالها يمكن برجة الحاسب وكتابة البرامج لحل مشاكل المستخدم ، هذه اللغة تسمى بلغة الآلة (Machine Language). وتتكون هذه اللغة من الرموز 0 و 1 حيث أن أوامر المعالج ما هي الا سلسلة معينة من هذه الرموز. فمثلاً لتعيين القيمة 31744 للمسجل AX^٢ يجب أن يحوي البرنامج على الأمر 101110001100000000000011. وبالتالي تكون عملية كتابة برنامج

^١ استحدثت عن معالجات إنتل ٣٢ بت في هذا البحث نظراً لأنها الأكثر انتشاراً.
^٢ المسجلات هي ذواكر بداخل المعالج.

متكامل بهذه اللغة أمراً في غاية الصعوبة وكذلك مهمة تنقيح البرنامج وتطويره في المستقبل هي معقدة أيضاً. لذلك ظهرت لغة التجميع لحل هذه المشكلة حيث أن اللغة تدعم مسميات ومختصرات للمسجلات ولأوامر المعالج ، فمثلاً الأمر السابق في لغة التجميع يكون بالصورة التالية.

Example ١.١ : Assembly Language

```
MOV AX, 0x7C00 ; Instead of 101110001100000000000111
```

والذي يجب تحويله الى لغة الآلة حتى يتمكن المعالج من تنفيذه ، هذا المحول يسمى بالمجمع والذي يقوم بتحويل أمر لغة التجميع الى ما يقابله بلغة الآلة^٣. ولم تنجح لغة التجميع في توفير لغة عالية المستوى تبسط عملية برمجة البرامج بشكل أكبر وذلك بسبب أنها مختصرات للغة الآلة لذلك سرعان ما تم تطوير لغات عالية المستوى مثل لغة السي^٤ والسي++ بحيث تكتب البرامج فيها بشكل مبسط بعيداً عن تعقيدات الآلة وأوامرها ومسجلاتها وتدعم هذه اللغات عدداً من التراكيب وجمل التحكم العالية المستوى. ولكي ينفذ المعالج برامج هذه اللغات فإنه يجب أولاً ترجمة الشفرة المصدرية الى ما يقابلها بلغة التجميع وهذا يتم عن طريق برنامج يسمى المترجم (Compiler) وبعدها يقوم المجمع بتحويل شفرة التجميع الى برنامجاً بلغة الآلة والذي يستطيع المعالج تنفيذه.

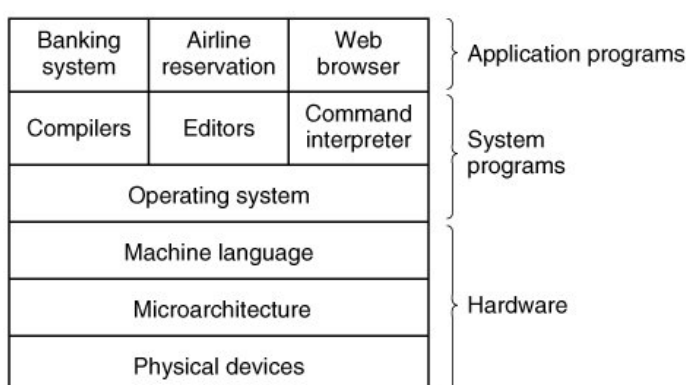
□ بخصوص تمثيل البيانات والبرامج في الحاسب فإنها تمثل بطرق مختلفة تختلف على حسب وحدة التخزين ولكنها في الآخر تستخدم المنطق الثنائي وهو وجود طاقة كهربائية أم لا ، فمثلاً تتكون الذاكرة الرئيسية DRAM من ملايين المكثفات (Capacitors) والترانزستورات (Transistors) لتكوين خلايا الذاكرة (Memory Cells) ، و تتكون كل خلية (والتي تشكل بت واحد من الذاكرة) من مكثف وترانزستور بحيث يحفظ المكثف قيمة الخلية (البت) والتي هي إما وجود إلكترون (منطقياً تساوي 1) وإما عدمها (منطقياً تساوي 0) ويعمل الترانزستور على تغيير قيمة المكثف . وعلى هذا الشكل تحفظ جميع الأوامر والبرامج في الذاكرة الرئيسية ويأتي دور المعالج لتنفيذ هذه الأوامر حيث يقوم بقراءتها وفهم وظيفتها (Decode) وتنفيذها ومن ثم يقوم بحفظ النتائج. ولكي ينفذ المعالج أي برنامج فإن البرنامج يجب أن يتواجد على الذاكرة الرئيسية وليس على أحد الدوائر الثانوية (مثل القرص الصلب).

حتى الان لم نذكر وظيفة نظام التشغيل لأن بيئة التشغيل الحقيقية هي المعالج وليست نظام التشغيل أو غيره من البرامج وعلى المبرمج الإلمام بكيفية برمجة عتاد ومتحكمات الحاسب وكيفية طباعة المخرجات على الشاشة وقراءة البيانات من متحكم لوحة المفاتيح ولا يقتصر على ذلك بل على المبرمج توفير طرقاً ودوالاً لإدارة الذاكرة من حجز المقاطع وتحريرها وكذلك إدارة جميع عتاد الحاسب. كل ذلك يجعل عملية كتابة البرامج مستحيلة وهذا ما أدى الى ظهور طبقة برمجية (Layer) تدير عتاد وموارد الحاسب

^٣ كل أمر بلغة التجميع يقابله أمراً واحداً بلغة الآلة لذلك حقيقة لا يوجد فرقاً في أداء البرامج المكتوبة بأي منهم ولا في حجم الملف الناتج ، وإنما يظهر الفرق في سهولة تطوير البرامج بلغة التجميع ولكن على حساب أنه يجب تحويلها عن طريق المجمع.
^٤ تم تطوير لغة السي بهدف برمجة نظام يونيكس Unix في معامل بيل.

وتوفر واجهة برمجية للمبرمج لكي يتعامل مع هذه الموارد. هذه الطبقة سميت **بنظام التشغيل (Operating System)**. الهدف الرئيسي لهذه الطبقة هي عزل المبرمج عن تعقيدات العتاد بحيث أن إدارة هذه العتاديات أصبحت من مهمة هذه الطبقة وفي نفس الوقت توفر واجهة برمجية (أو جهاز تخيلي) للاستفادة من هذه العتاديات. والشكل ٢.١ يوضح موضع هذه الطبقة (نظام التشغيل) في حالة قسمنا جهاز الحاسب إلى عدة طبقات [?]. وأدنى طبقة هي طبقة العتاديات (Device Level) حيث تتكون من المتحكمات

شكل ٢.١: طبقات الحاسب



ومن الشرائح المتكاملة (Integrated Circuit) والأسلاك وكل ما يتعلق بالأجهزة المادية. يلي هذه الطبقة طبقة Microarchitecture وفيها تظهر برمجيات (Microprogram) تتحكم في عمل المتحكمات لكي تؤدي وظيفتها فمثلاً برمج ال data path بداخل المعالج والذي يقوم في كل دورة للساعة (Clock Cycle) بجلب قيمتين من المسجلات إلى وحدة الحساب والمنطق (Arithmetic Logic Unit) التي تجري عليهم عملية ما ومن ثم تقوم بحفظ النتيجة في أحد المسجلات. وظيفة data path هي تنفيذ الأوامر والتعليمات وذلك بارسالها إلى وحدة الحساب والمنطق، وتشكل مجموعة الأوامر المدعومة وكذلك المسجلات المرئية للمبرمج لغة التجميع طبقة مجموعة الأوامر (Instruction Set Architecture) وتسمى هذه الطبقة طبقة الآلة (Machine Language) حيث تحوي على كل الأوامر التي يدعمها المعالج بما فيها أوامر القراءة والكتابة من مسجلات متحكمات العتاد (Device Controller). يلي هذه الطبقة طبقة نظام التشغيل والتي تفصل وتعزل العتاد عن المستخدم فبدلاً من أن يقوم المبرمج ببرمجة متحكم القرص الصلب ونظام للملفات حتى يتمكن من قراءة ملف على القرص فإن النظام يوفر واجهة مبسطة بالصورة `read(fd, buffer, size)`. وأخيراً توجد طبقة البرامج (برامج النظام والمستخدم) ولا تصنف الكثير من برامج النظام ضمن نظام التشغيل حيث أن البرامج التي تتبع لنظام التشغيل يجب أن تعمل في

مستوى النواة (Kernel Mode) وليس في المستويات الأخرى^٥.

١.١ ما هو نظام التشغيل

من الصعب إيجاد تعريفاً واضحاً لأنظمة التشغيل فما يعتبره البعض تابعاً لنظام ما لا يعتبره الآخرون كذلك. لكن ما تم الاتفاق عليه هو أن **نظام التشغيل** يدير عتاد وموارد الحاسب ويوفر واجهة برمجية (جهاز تخيلي) من خلالها يمكن الاستفادة من هذه الموارد.

١.١.١ نظام التشغيل كجهاز تخيلي

مما سبق نجد أن الواجهة التي تقدمها طبقة الآلة (Machine Language Level) هي بدائية ويصعب استخدامها في كتابة البرامج ، فكما ذكرنا كمثال للقراءة من ملف على القرص يجب أن يحوي البرنامج على شفرة لنظام الملفات حتى نعرف عنوان الملف الفيزيائي على القرص، وكذلك يجب أن يحوي البرنامج على شفرة للتعامل مع متحكم القرص الصلب وهي شفرة ليست باليسيرة حيث للقراءة من القرص يجب تحديد رقم المقطع ورقم الرأس ورقم المسار وتحديد الذاكرة المؤقتة (Buffer) حتى يتم تحميل المقاطع إليها. كل هذه الأمور لو استمرت بهذا الشكل لما وصلت التطبيقات لما هي عليها الان، لذلك كان الحل هو بإيجاد طبقة نظام التشغيل والتي توفر واجهة أو أوامر مبسطة وبمجردة من تفاصيل وتعقيدات العتاد لكي تستخدمها البرامج بدلا من الأوامر التي توفرها طبقة الآلة .

٢.١.١ نظام التشغيل كمدير للموارد والعتاد

بعد أن تم عزل المبرمج بواسطة طبقة نظام التشغيل فان هذه الطبقة تقدم بجانب الواجهة البرمجية إدارة لعتاد الحاسب (المعالج،الذاكرة،الأقراص الصلبة والمرنة، كرت الشبكة، وغيرها من المتحكمات) ، ومهمة إدارة العتاد تتركز في حجز العتاد وتحريره ، فمثلاً يقوم نظام التشغيل بإدارة المعالج نفسه وذلك بأن يحجز المعالج لبرنامج ما ومن ثم يحجز المعالج ويحجزه لبرنامج آخر (تعدد المهام Multitasking) وكذلك يدير النظام أهم موارد الحاسب وهي الذاكرة الرئيسية وذلك بحجز مقاطع من الذاكرة (Memory Blocks) بناءاً على طلب برامج المستخدم وكذلك عملية تحرير الذواكر وإدارة الذاكرة التخيلية ومفهوم الصفحات.

^٥ في الفصل الثاني بإذن الله سيتم الحديث عن مستويات الحماية في المعالجات.

٢.١ تاريخ أنظمة التشغيل

خلال سنوات مضت تطورت أنظمة التشغيل تطوراً ملحوظاً من أنظمة تشغيل لبرنامجاً واحداً الى أنظمة موزعة تسمح بتشغيل أكثر من برنامج على عدة حواسيب مختلفة. هذا التطور سببه الرئيسي تطور الحاسبات والمعالجات وازدياد حجم الذواكر بشكل رهيب. وفي هذا الجزء سنتلقي نظرة على تطور أجيال الحواسيب وبعض أنظمة التشغيل التي استخدمت في تلك الفترات.

١.٢.١ الجيل الصفري (من ١٦٢٤ حتى ١٩٤٥): الحواسيب الميكانيكية

أول محاولة لبناء آلة حسابية كانت من قبل العالم الفرنسي بليز باسكال^٦ في عام ١٦٤٢ عندما كان عمره ١٩ عاماً وذلك لمساعدة والده الذي كان يعمل محصلاً للضرائب لمصلحة الحكومة الفرنسية. هذه الآلة (وتعرف بالاسم Pascaline) هي ميكانيكية بالكامل وتوفر فقط عملية الجمع والطرح (انظر الشكل ٣.١).

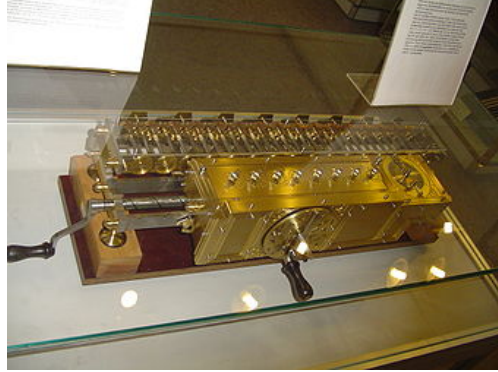
شكل ٣.١: آلة باسكال



وبعد حوالي ٣٠ عاماً قام العالم الرياضي جوتفريد ليبتر Gottfried Wilhelm Leibniz ببناء آلة حسابية ميكانيكية أخرى (تم الإنتهاء منها في عام ١٦٩٤ وسميت بالاسم Step Reckoner) ولكن هذه المرة أصبح من الممكن إجراء العمليات الحسابية الأربعة: الجمع والطرح والضرب والقسمة (الشكل ٤.١). ومضت حوالي ١٥٠ عاماً بدون أي شيء يذكر حتى قام البروفيسور شارلز باباج Charles Babbage بتصميم آلة محرك الفروق Difference engine (انظر الشكل ٥.١)، وهي آلة ميكانيكية أيضاً تشابه آلة باسكال في أنها لا توفر سوى عمليتي الجمع والطرح لكن هذه الآلة تم تصميمها لغرض حساب قيم دوال كثيرات الحدود باستخدام طرق التقريب المنتهية (Method of Finite Differences). وما ميز هذه الآلة هي طريقة إخراج النتائج حيث تنقش النتائج على ألواح نحاسية. وعلى الرغم من أن آلة الفروق عملت جيداً

^٦والذي تم تسمية لغة البرمجة باسكال باسمه تشريفاً له.

شكل ٤.١: آلة Step Reckoner في متحف بألمانيا



شكل ٥.١: محرك الفروق بعد أن قام ابن بابايح بتجميعه



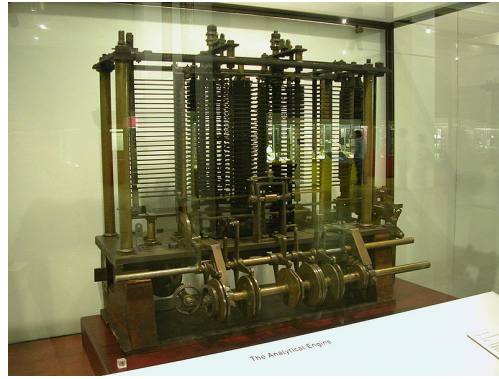
إلا أن تصميمها كان يسمح بحساب خوارزمية واحدة فقط^٧ وهذا ما جعل شارلز بابايح يعيد محاولته مجدداً ويستهلك جزءاً ضخماً من وقته ومن ثروة حكومته في بناء آلة أخرى عُرفت بالمحرك التحليلي Analytical Engine (انظر الشكل ٦.١). هذا المحرك (وهو أيضاً آلة ميكانيكية بالكامل) احتوى على أربع مكونات: المخزن (الذاكرة Memory)، الطاحنة (وحدة الحساب Computation Unit)، وحدة الإدخال (قارئ البطاقات المثقبة Punched Card Reader) ووحدة الإخراج (البطاقات المثقبة واللوحات المطبوعة). ويتكون المخزن من ١٠٠٠ كلمة (Word) بطول ٥٠ رقم صحيح وتستخدم لحفظ المتغيرات والنتائج، أما الطاحنة فتستقبل الوسائط من المخزن وتجري عليهم أي من العمليات الرياضية الأربعة ومن ثم تحفظ النتائج في المخزن. الميزة الأساسية للمحرك التحليلي هو قدرته على حل عدد كبير من المشاكل

^٧ في عام ١٩٩١ قام متحف العلوم بلندن ببناء نموذج مكتمل لمحرك الفروق.

٢.١ تاريخ أنظمة التشغيل

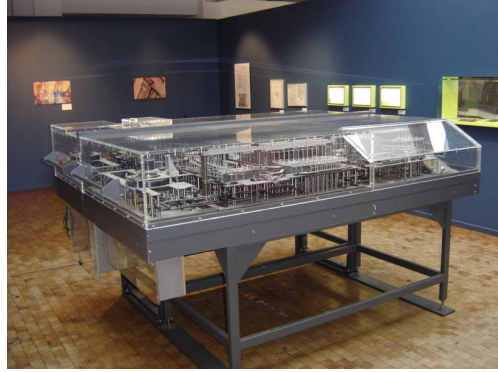
(على عكس محرك الفروق) حيث تكتب البرامج في بطاقات مثقبة ويتم قرائتها إلى المحرك بواسطة قارئاً لهذه البطاقات. وتحتوي هذه البطاقات على أوامر موجهة إلى المحرك لكي يقوم بقراءة عددين من المخزن ويجري عملية ما (جمع مثلاً) ومن ثم يحفظ النتيجة في المخزن أيضاً ، وكذلك تحوي أوامر أخرى مثل المقارنة بين عددين والتفرع ونقل التنفيذ. ولأن المحرك قابل للبرمجة (بلغة شبيهة بلغة التجميع) فقد استعان شارلز باباج بالبرمجة آدا لوفلاس Ada Lovelace والتي صنفت كأول مبرمج في التاريخ. ولسوء الحظ لم ينجح شارلز باباج في أن يزيد من دقة المحرك ربما لأنه يحتاج إلى آلاف من التروس والعجلات. وبشكل أو بآخر يعتبر شارلز باباج الجد الأول للحواسيب الحالية حيث أن فكرة عمل المحرك التحليلي مشابهة للحواسيب الحالية.

شكل ٦.١: المحرك التحليلي.متحف في لندن

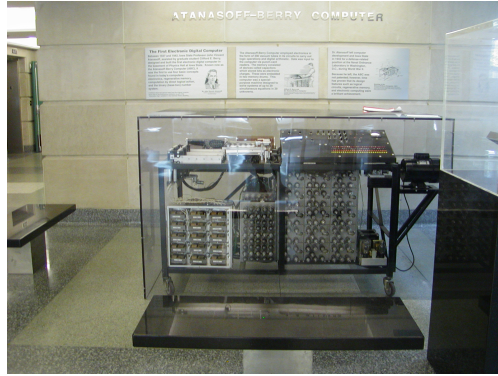


وفي أواخر ١٩٣٠ قام الطالب الألماني كونارت تسوزا Konrad Zuse ببناء آلة حسابية ولكنها تعتمد على الريلاي (Relay) وسميت بجهاز Z1 (انظر الشكل ٧.١) وتعتبر أول حاسبة تعتمد على الريلاي وعلى المنطق الثنائي في عملها. ولسوء الحظ تم تدمير الحاسبة Z1 في انفجار في برلين أثناء الحرب العالمية الثانية عام ١٩٤٣. وعلى الرغم من أن تصميم Z1 لم يؤثر في تصميم الحواسيب التي تليه بسبب تدميره هو وجميع خطط بنائه إلا أنه يعتبر أحد التصميمات التي كان لها أثرها ذاك الوقت. وبعد برهة من الزمن قام جون أتاناسوف John Vincent Atanasoff بتصميم جهاز Atanasoff (انظر الشكل ??) والذي كان مدهشاً في وقته حيث يستخدم التحسب الثنائي (Binray Arithmetic) ويجوي مكثفات للذاكرة ولكن الجهاز لم يكن عملياً. وفي بدايات ١٩٤٠ قام هوارد ايكن Howard Aiken بتصميم الحاسبة ASCC - والتي أعيد تسميتها إلى Harvard Mark I - في جامعة هارفارد (انظر الشكل ٩.١). وقد أتبع هوارد نفس منهج باباج وقرر أن يعتمد على الريلاي وأن يصمم حاسبة للأغراض العامة والتي فشل بها شارلز باباج. وفي عام ١٩٤٤ تم الإنتهاء من تصميمها وتم تصميم نسخة محسنة أيضاً سميت ب Harvard Mark II. ومع هذا التصميم انتهى عصر الحاسبات الميكانيكية والريلاي وبدأ عصر جديد.

شكل ٧.١: حاسبة Z1 بعد إعادة انشائها في متحف بألمانيا

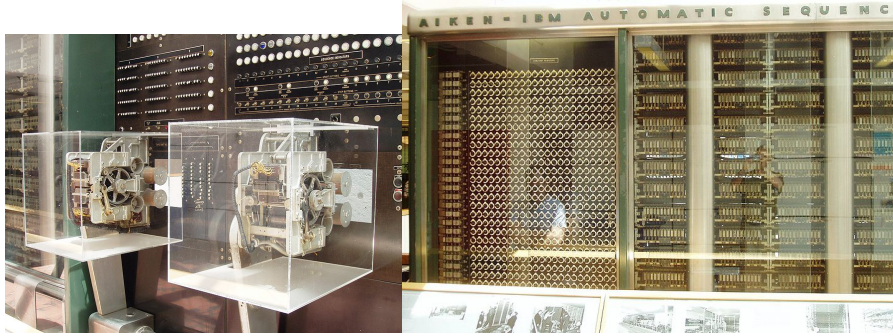


شكل ٨.١: حاسبة Atanasoff بعد إعادة انشائها في جامعة Iowa State



٢.٢.١ الجيل الأول (من ١٩٤٥ حتى ١٩٥٥): الصمامات المفرغة و لوحات التوصيل

في بدايات الحرب العالمية الثانية ، كان أمير البحرية في برلين يرسل رسائل الى الغواصات الألمانية عبر موجات الراديو والتي استطاع حلفاؤها البريطانيون التقاطها ، ولسوء حظهم كانت الرسائل ترسل مشفرة وذلك عن طريق شفرة خاصة تنتج من قبل جهاز يسمى بجهاز إنجما (Enigma Machine) والذي تم تصنيعه لتشفير الرسائل وفك تشفيرها . وقد تمكنت الإستخبارات البريطانية من الحصول على أحد هذه الأجهزة وذلك بعد الإتفاق مع الإستخبارات البولندية والتي كانت قد سرقت جهازاً من الألمان. وحتى يتمكن البريطانيون من فك شفرة الرسائل فان هذا يتطلب وقتاً وعمليات حسابية طويلة ، لذلك سرعان ما أسست معملاً سرياً يحوي على حاسبة الكترونية عرفت بالاسم Colossus. هذه الحاسبة تم تصميمها



(ب) الإدخال والإخراج والتحكم

(أ) الجزء الأيسر من حاسبة Mark I

شكل ٩.١: حاسبة Harvard Mark I

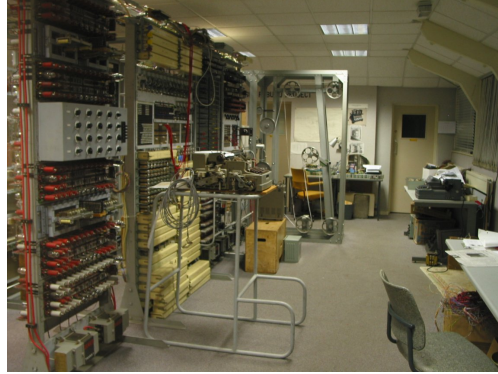
شكل ١٠.١: آلة إنجما الألمانية لتشفير الرسائل وفكها



من قبل عدة أشخاص وشارك فيها العالم آلان تورنج وأصبحت جاهزة للعمل في عام ١٩٤٣. ويسبب كون المشروع سريا وتم التكتّم عليه لما يقارب ٣٠ عاما فإن هذا النوع من الحواسيب لم يؤثر على تصاميم

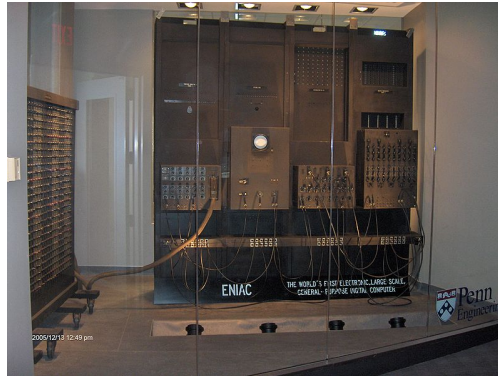
الحواسيب الحديثة ولكن يجدر بالذكر أن هذه الحاسبة تعتبر أول حاسبة إلكترونية قابلة للبرمجة تستخدم الصمامات الهوائية في حساباتها.

شكل ١١.١: الحاسبة colossus التي كسرت شفرة إنجما



وفي عام ١٩٤٣ قدم جون موكلي John Mauchley مقترحاً الى الجيش الأمريكي طالباً تمويله بالمال للبدء بتصميم حاسبة إلكترونية لحساب جداول إطلاق المدفعية بدلاً من حسابها يدوياً وذلك لتقليل الأخطاء وكسب الوقت، وقد تمت الموافقة على المشروع وبدأ جون موكلي وطالبه الخريج إيكريت ببناء حاسبة تم تسميتها بالاسم إيناك ENIAC اختصاراً للجملة Electronic Numerical Integrator And Computer. وتتكون من ١٨٠٠٠ صماماً مفرغاً (Vacuum Tubes) و ١٥٠٠ حاكمة (Relays)، وتزن الحاسبة ٣٠ طن وتستهلك ١٤٠ كيلو واط من الطاقة. وداخليا تحتوي الحاسبة على ٢٠ مسجل كل منهم يسع عدداً صحيحاً بطول ١٠ خانات. وتتم برمجة إيناك عن طريق ٦٠٠٠ مفتاح Switch.

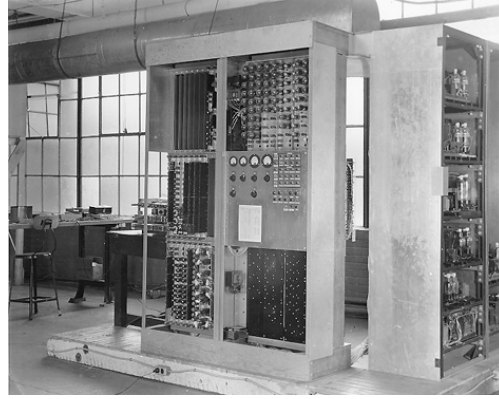
شكل ١٢.١: الحاسبة ENIAC



٢.١ تاريخ أنظمة التشغيل

وقد تم الإنتهاء من تصميم إيناك عام ١٩٤٦ ، الوقت الذي كانت الحرب قد انتهت ولم تستخدم الحاسبة لهدفها الرئيسي. وبعد ذلك نظم جون موكلي وطالبه إيكريت مدرسة صيفية لوصف مشروعاتهم للباحثين والمهتمين. الأمر الذي أسفر عن ظهور عدد كبير من الحاسبات الضخمة . وأول حاسبة بعدها كانت EDSAC في عام ١٩٤٩ بواسطة ويلكس في جامعة كامبردج. كذلك تلتها الحاسبات JOHNIAC و ILLIAC وغيرهم. بعد ذلك بدأ جون موكلي وإيكريت بالعمل على حاسبة أخرى سميت بالاسم EDVAC اختصاراً للجملة Electronic Discrete Variable Automatic Computer والتي كانت تعمل بالأرقام الثنائية بدلاً من العشرية (كما في إيناك). بعد ذلك توقف المشروع بسبب أن جون وإيكريت قد أنشأ شركتهم الخاصة.

شكل ١٣.١: الحاسبة EDVAC



٣.٢.١ الجيل الثاني: الترانزستورات

٤.٢.١ الجيل الثالث: الدوائر المتكاملة

٥.٢.١ الجيل الرابع: الحواسيب الشخصية